DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/98M-3

Parameter Estimation of the Mixed Generalized Gamma Distribution Using Maximum Likelihood Estimation and Minimum Distance Estimation

THESIS

Dean G. Boerrigter, Capt. USAF

AFIT/GOR/ENS/98M-3

AFIT/GOR/ENS/98M-3

PARAMETER ESTIMATION OF THE MIXED GENERALIZED GAMMA DISTRIBUTION USING MAXIMUM LIKELIHOOD ESTIMATION AND MINIMUM DISTANCE ESTIMATION

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Dean G. Boerrigter

Captain, USAF

March 1998

Approved for public release, distribution unlimited

AFIT/GOR/ENS/98M-3

PARAMETER ESTIMATION OF THE MIXED
GENERALIZED GAMMA DISTRIBUTION
USING MAXIMUM LIKELIHOOD ESTIMATION
AND
MINIMUM DISTANCE ESTIMATION

Dean G. Boerrigter

Captain, USAF

Approved:

Advisor: Maj. Edward A. Pohl, Ph.D.	11 MAR 98 date
Co-Advisor: Maj. W. Paul Murdock, Jr., Ph.D.	11 MAR 98 date
Reader: Dr. Albert C. Moore, Ph.D.	11 Mar 98 date

Acknowledgments

I would first like to thank Jesus Christ for the opportunity to try to glorify Him with this contribution to the field. For the Greater Glory of God. Acts 4:8-12.

I would like to thank the members of my committee Major Pohl, Major Murdock, and Dr. Moore for all the time and expertise that they gave supporting me in this thesis effort. I'd also like to thank Maj Reid who answered a numerical question or two that I had.

I would like to thank Audra Cake for her help with FORTRAN and reteaching me some of the basic properties of logarithms, Jim Cordeiro for his help checking my calculus, and especially Alan & Suzy Hulsopple, Carri Gray and many others for their prayers for my effort. I would like to thank Jim Cordeiro and Mark Friend for letting me use their dedicated computer (I always let them borrow it when they asked), and Jim Look for performing many replications for me on his computer. I would also like to thank Mike Warner for the replications he ran for me and for the input he gave on my write up and defense presentation. Lastly, I would like to thank Steve Oimen for his help putting together my thesis defense.

I will finish with mentioning that although I used Genetic Algorithms that are based in Darwinian evolutionary theory, I in no way endorse the theory. The preponderance of scientific evidence is against it. Genetic Algorithms work because they are an artificial representation of the theory. Johnston's Darwin on Trial shows that Darwinian evolutionary theories are based only on presupposition and blind faith in science.

Dean Boerrigter

Table of Contents

Page

List of Tables

List of Figures

List of Terms

Ave	Average
CDF	Cumulative Distribution Function
Dist	Integrated Squared Distance
GA	Genetic Algorithm
GGD	Generalized Gamma Distribution
GGD4	Four-Parameter Generalized Gamma Distribution
GGD9	Nine-Parameter Mixed Generalized Gamma Distribution
MDE	Minimum Distance Estimation
mGA	Micro-Genetic Algorithm
MLE	Maximum Likelihood Estimation
MSE	Mean Squared Error
PDF	Probability Distribution Function
StDev	Standard Deviation

Abstract

The Generalized Gamma is an extremely flexible distribution that is useful for reliability modeling. Among its many special cases are the Weibull and Exponential distributions. A mixture of Generalized Gamma Distributions is even more useful because multiple causes of failure can be simultaneously modeled.

This research studied parameter estimation of the special cases of the Mixed Generalized Gamma Distribution and built upon them until the full nine-parameter distribution was being estimated. First, special cases of a single Generalized Gamma Distribution were estimated. Next, mixtures of Exponential distributions with both known and unknown location parameters were estimated. Next, mixtures of Weibull distributions with both known and unknown location parameters were estimated. Lastly, the full nine-parameter Mixed Generalized Gamma Distribution was estimated.

Two techniques were used to estimate the parameters of each distribution. The first technique used was the Method of Maximum Likelihood. The log likelihood equation was maximized using a Genetic Algorithm. The second technique used was the Method of Minimum Distance. This technique takes the Maximum Likelihood parameter estimate as initial estimate. With this initial estimate, the mixture and the first location parameter are sequentially varied to minimize the Anderson-Darling statistic between the estimated cumulative distribution function and the empirical distribution function. These two parameters are then fixed at their Minimum Distance values and the remaining parameters are re-estimated using Maximum Likelihood.

Minimum Distance Estimation was demonstrated to improve the parameter estimates from Maximum Likelihood for almost all of the special case distributions tested. It did not improve the estimate for the full nine-parameter Mixed Generalized Gamma Distribution, but this was because the technique used to find the Maximum Likelihood parameter estimates performed poorly and did not return a good initial estimate for Minimum Distance.

PARAMETER ESTIMATION OF THE MIXED GENERALIZED GAMMA DISTRIBUTION USING MAXIMUM LIKELIHOOD ESTIMATION AND MINIMUM DISTANCE ESTIMATION

I. Introduction

1.1 Background

Weapon systems and support systems are becoming more complex and are also becoming more expensive. In an era of shrinking defense budgets, the reliability of these systems become paramount. Reliability analysis is used to obtain the probability of a component's ability to perform a given mission (26:1). Probability distributions are used to model failure times. The better the probability distribution fits the sample data, the more likely it is to predict failure of the component or system being modeled. Accurate failure models can save money by "right-sizing" maintenance structure. This could cause one of two possible problems. Inaccurate models can mean either unneeded and expensive maintenance capability could be bought. On the other hand, not enough maintenance could be acquired, which would degrade operational readiness. Typical probability distributions used in modeling failure data include the Exponential and the Weibull distributions. Embedding these competing distributions into a single parametric framework would allow a comprehensive test to determine which model provided the better functional form to model failure times (12:69). Multiple distributions could be tested at once and then the results could then be checked to see if they match any of the special cases. One candidate distribution is the Generalized Gamma Distribution. Special cases of it include the Half-Normal, Exponential, Gamma, Weibull and the Chi Squared Distributions (51:351). It can show four models of hazard functions—bathtub, inverted bathtub, increasing and decreasing. It is the only distribution capable of showing all four types with the selection of the proper parameters (47:280). The Mixed Generalized Gamma Distribution can be used to model components that have two causes of failure, such as sudden catastrophic failures and wear-out failures (43:1799). It is a highly flexible distribution, but there has been some reluctance to use it because of the difficulty involved in estimating its parameters.

1.2 Problem Statement

The formal statement of the problem is compare the parameter estimation techniques of Maximum Likelihood Estimation and Minimum Distance Estimation for the Mixed Generalized Gamma Distribution to determine if Minimum Distance gives better parameter estimates than Maximum Likelihood Estimation alone. The closer the estimated parameters are to the true parameters, the better the distribution fit will be and the more accurate information the distribution will give. Two methods will be tested to estimate the parameters from random variates generated from a Mixed Generalized Gamma distribution. One is the Method of Maximum Likelihood, which estimates parameters by maximizing the log likelihood equation. A second method is called the Method of Minimum Distance, which starts with the MLE parameters and then iteratively adjusts the parameter estimates of the cumulative distribution function against the empirical distribution function (EDF) in order to improve them. This method has been shown to improve parameter estimation for distributions such as the three-parameter Generalized Gamma by William James in 1980 (26), the four-parameter Generalized Gamma distribution by Shumaker in 1982 (49), the mixture of Exponential distributions by Benton-Santo (1) and the Mixed Weibull Distribution by Donald Mumford in 1996 (37). It is therefore believed that the Method of Minimum Distance will improve parameter estimates for the Mixed Generalized Gamma Distribution.

II. Literature Review

Five topics will be discussed in the chapter: the Generalized Gamma Distribution and its parameter estimation, Genetic Algorithms, Maximum Likelihood Estimation, Minimum Distance Estimation, and Random Variate Generation. Samples of random variates were generated from Generalized Gamma Distribution. Two parameter estimation techniques were employed to calculate sample parameter estimations: Maximum Likelihood Estimation and Minimum Distance Estimation. A Genetic Algorithm was used to maximize the result of the maximum likelihood equation used in both estimation techniques.

2.1 Generalized Gamma Distribution

The probability distribution function of the Generalized Gamma Distribution (GGD) is given by:

$$f(x;c,a,b,p) = \frac{p\cdot(x-c)^{b\cdot p-1}\cdot e^{-[(x-c)/a]^p}}{a^{b\cdot p}\cdot\Gamma(b)}$$

where a, b, p ≥ 0 and x $\geq$ c $\geq$ 0 (20:2).

The function has four parameters: c is the location parameter, a is the scale parameter, b is the shape/power parameter, and p is the power parameter. $\Gamma(z)$ is the Gamma function, defined by

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt \quad (31{:}332).$$

The cumulative distribution function of the Generalized Gamma Distribution is given by (20:2):

$$F(x;c,a,b,p) = \frac{\Gamma_{[(x-c)/a]^p}(b)}{\Gamma(b)}$$

The numerator is the Incomplete Gamma Function, which means that it is the Gamma function integrated to a finite number, instead of to infinity (32:512-513). The Incomplete Gamma Function $\Gamma_X(z)$ is defined by

$$\Gamma_X(z) = \int_0^X t^{z-1}e^{-t}dt.$$

The finite number in this case is $[(x-c)/a]^p$. The numerator is normalized by the denominator, thus letting the function range from 0 to 1, which is a necessary condition for the definition of a cumulative distribution

function. Since the cumulative distribution function is an incomplete Gamma function ratio, this suggests the name Generalized Gamma (20:2).

The partial derivatives of the four parameter Generalized Gamma Distribution with respect to each of the parameters may be defined with the aid of the following two auxiliary functions:

$$S = \left(\frac{x-c}{a}\right)^{p}$$

$$T = (x-c)^{b\cdot p-1} \cdot \frac{\exp(-S)}{a^{b\cdot p} \cdot \Gamma(b)}$$

The partial derivatives with respect to the parameters c, a, b and p are:

$$\partial f / \partial c = \frac{T}{x-c} \cdot \{-p \cdot (b \cdot p - 1) + p^2 \cdot S\}$$

$$\partial f / \partial a = \frac{p^2}{a} \cdot T \cdot \{S - b\}$$

$$\partial f / \partial b = T \cdot \{p^2 \cdot \ln(x-c) - p^2 \cdot \ln(a) - p \cdot \Psi(b)\}$$

$$\partial f / \partial p = T(x,c,a,b,p) \cdot \{1 + p \cdot b \cdot \ln(x-c) - p \cdot S \cdot \ln(\tfrac{x-c}{a}) - p \cdot b \cdot \ln(a)\}$$

Table 1 First Derivatives of the Generalized Gamma Distribution

where $\Psi(x)$ is the Psi or Digamma function, which is defined as $\Psi(x) \equiv \dfrac{d \ln\{\Gamma(x)\}}{d x} \quad x > 0$ (32:513,3,53).

2.1.1 A Brief History of Generalized Gamma Distribution Development

The three-parameter Generalized Gamma Distribution (GGD) was originally proposed by E.W. Stacy in 1962 (51,10:423). His three-parameter GGD is equivalent to the four-parameter GGD with the location parameter, c, set equal to zero. The generalization was accomplished by supplying a positive parameter as an exponent in the exponential factor of the gamma distribution (51:1187). In 1965, Parr and Webster demonstrated the usefulness of the distribution as a model for failure density function in reliability

predictions (39:1). Harter made the GGD more general by adding a location parameter. Harter wanted to enhance the usefulness of the Generalized Gamma Distribution for reliability modeling (21:159). In 1966, Harter developed an iterative procedure for the Maximum Likelihood Estimates of Generalized Gamma Distribution parameters and the asymptotic variances and covariance of the maximum likelihood estimators for complete and censored samples (20:7-9). Stacy and Mihram demonstrated a further generalization by including cases where the power parameter, p, could be negative (52:349). In 1970s, Hager and Bain developed inferential procedures for the three parameter GGD, and compared its reliability estimates with the Weibull (18:1601, 19:547). In 1980, Hobbs, Moore and James used Minimum Distance to estimate the parameters for the three-parameter Generalized Gamma Distribution (26:vi, 22:237). In 1982, Shumaker used Minimum Distance to estimate the parameters for the four-parameter Generalized Gamma Distribution (49:22). In 1987, Wingo developed a method to find the maximum likelihood parameter estimates for the three-parameter GGD using numerical root isolation because of the numerical difficulties that can occur fitting the GGD parameters (64:586). In 1991, Rao, Kantam and Narasinkham developed estimators for the location and scale parameters for the GGD (44:3823). In 1995, Pham and Almhana presented the hazard rate for the three parameter GGD (42:392).

In reliability and life testing, several distributions are often used to model failure times. It has been suggested by Farewell and Prentice that embedding these competing distributions into a single parametric framework would allow running a comprehensive test to compare them (12:69). The Generalized Gamma contains a number of important distributions as special cases. Some examples of special cases as shown by Stacy and Mihram are listed in Table 2 (52:351). Typical shapes for the PDF are given in Figure 1.

Table 2 Special Cases (a, b, p>0) of the GGD4

GGD4(c,a,b,p)	Equivalent Distribution
(0,B,1,1)	Exponential (B)
(0,B,α,1)	Gamma(α,B)
(0,B,1,α)	Weibull(α,B)
(0,$\sqrt{2}$,n/2, 1)	Chi Squared(n)
(0, $\sqrt{2}$,1/2,2)	Half Normal
(0,c$\sqrt{2}$,1,2) c>0	Rayleigh

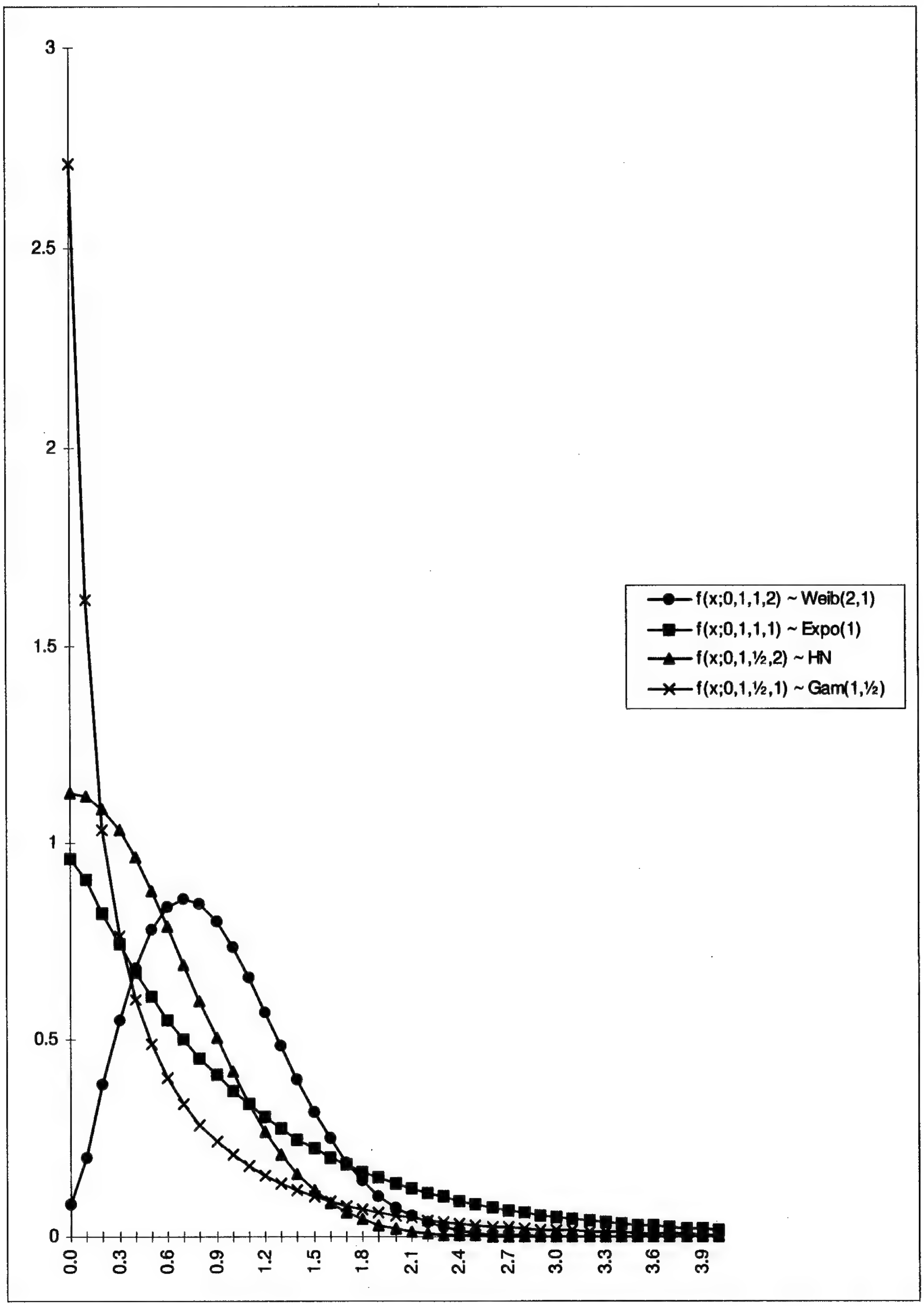

Figure 1 Typical Shapes for f(x; c, a, b, p)

2.1.2 A Selected History

Parr and Webster studied the Generalized Gamma distribution to discriminate between failure density functions, particularly the Weibull and the Exponential, which are both special cases of the Generalized Gamma (39). The Maximum Likelihood Equations that they derived for the three-parameter Generalized Gamma distribution with known location are for the following estimated parameters:

$$a = \left[\left(\frac{p}{n \cdot d} \right) \cdot \sum_{i=1}^{n} x_i^p \right]^p$$

$$\ln(p) - \ln(n) - \ln(d) + \ln\left(\sum_{i=1}^{n} x_i^p \right) - \left(\frac{p}{n} \right) \cdot \sum_{i=1}^{n} \ln(x_i) = 0$$

$$\ln(p) - \ln(n) - \ln(d) + \ln(\sum_{i=1}^{n} x_i^p) + \left(\frac{p}{d} \right) - \frac{p}{\sum_{i=1}^{n} x_i^p} \cdot \sum_{i=1}^{n} \left(x_i^p \cdot \ln(x_i) \right) = 0$$

where $d = b \cdot p$ and $b, p > 0$ and n is the sample size.

Radhakrishna, Rao and Anjaneyulu derived the moment and maximum likelihood estimators for mixture of a pair of three-parameter Generalized Gamma Distributions in order to study catastrophic and wear-out failure life-testing data (43). Their three-parameter Generalized Gamma Distribution is the same as the four parameter distribution with the location parameter set to zero:

$$\mathrm{f(x;a,b,p)} = \frac{\mathrm{p} \cdot \mathrm{x}^{\mathrm{b \cdot p - 1}} \cdot \mathrm{e}^{-[\mathrm{x/a}]^{\mathrm{p}}}}{\mathrm{a}^{\mathrm{b \cdot p}} \cdot \Gamma(\mathrm{b})}$$

and their mixture is defined as

$$g(x;a_1,b_1,p_1, a_2,b_2,p_2,m)=m * f(x; a_1,b_1,p_1) + (1-m) * f(x; a_2,b_2,p_2).$$

Thus their log likelihood function is

$$LL = \sum_{i=1}^{n} \ln\{g(x;a_1,b_1,p_1, a_2,b_2,p_2,m)\}.$$

The maximum likelihood equations to be solved are defined as:

$$\frac{dLL}{d\theta_j} = \sum_{i=1}^{n} \frac{1}{g(x_i)} \cdot \frac{dg}{d\theta_j}$$

where θ_j is each of the seven parameters. Thus, to obtain the Maximum Likelihood Estimates, these equations are to be solved simultaneously using a numerical technique such as Newton-Raphson or the method of steepest descent.

2.1.3 Mixed Generalized Gamma Distribution

In reliability studies, there can be more than one cause for failure in a population of components. Attempting to fit a unimodal distribution to account for two separate causes may not fit either type well, particularly if the failure times associated with both are widely separated. One method of working with multiple causes of failure is to use a mixture distribution. A mixture distribution is a distribution made of one or more component distributions. Its probability distribution function is of the form:

$$p(x) = m_1 f_1(x) + \ldots + m_n f_n(x) \qquad \text{where } \Sigma m_i = 1$$

m_i is the probability of being from the component distribution i, $f_i(x)$ is the probability distribution function of component distribution i, and n is the number of component distributions being mixed (61:1). In the case where only two component distributions exist, the parameters can be defined simply as m and $(1 - m)$.

The Mixed Generalized Gamma Distribution considered in this thesis is a bimodal mix of the Generalized Gamma Distribution. The probability density function for the Mixed Generalized Gamma Distribution is as follows:

$$f(x) = m \cdot \left(\frac{p_1 \cdot (x-c_1)^{b_1 \cdot p_1 - 1} e^{-[(x-c_1)/a_1]^{p_2}}}{a^{b_1 \cdot p_1} \cdot \Gamma(b_1)} \right) + (1-m) \cdot \left(\frac{p_2 \cdot (x-c_2)^{b_2 \cdot p_2 - 1} e^{-[(x-c_2)/a_2]^{p_2}}}{a^{b_2 \cdot p_2} \cdot \Gamma(b_2)} \right)$$

where $a_1, a_2, b_1, b_2, p_1, p_2 > 0$, $0<m<1$ and $x \geq c_2 \geq c_1 \geq 0$.

In this equation c_1 and c_2 are the location parameters, a_1 and a_2 are scale parameters, b_1 and b_2 are shape/power parameters, p_1 and p_2 are power parameters, while m is the mixture parameter.

As an example of a Mixed Generalized Gamma Distribution probability distribution function, Figure 2 contains three functions: g(x) = .5 f1(x) + .5 f2(x) where

f1(x) = GGD4(0,2,1,1) f2(x) = GGD4(10,2,1,4)

These are equivalent to:

f1(x)= Weibull(1,2) f2(x) = Weibull(4,2)+10 = Weibull(4,2,10)

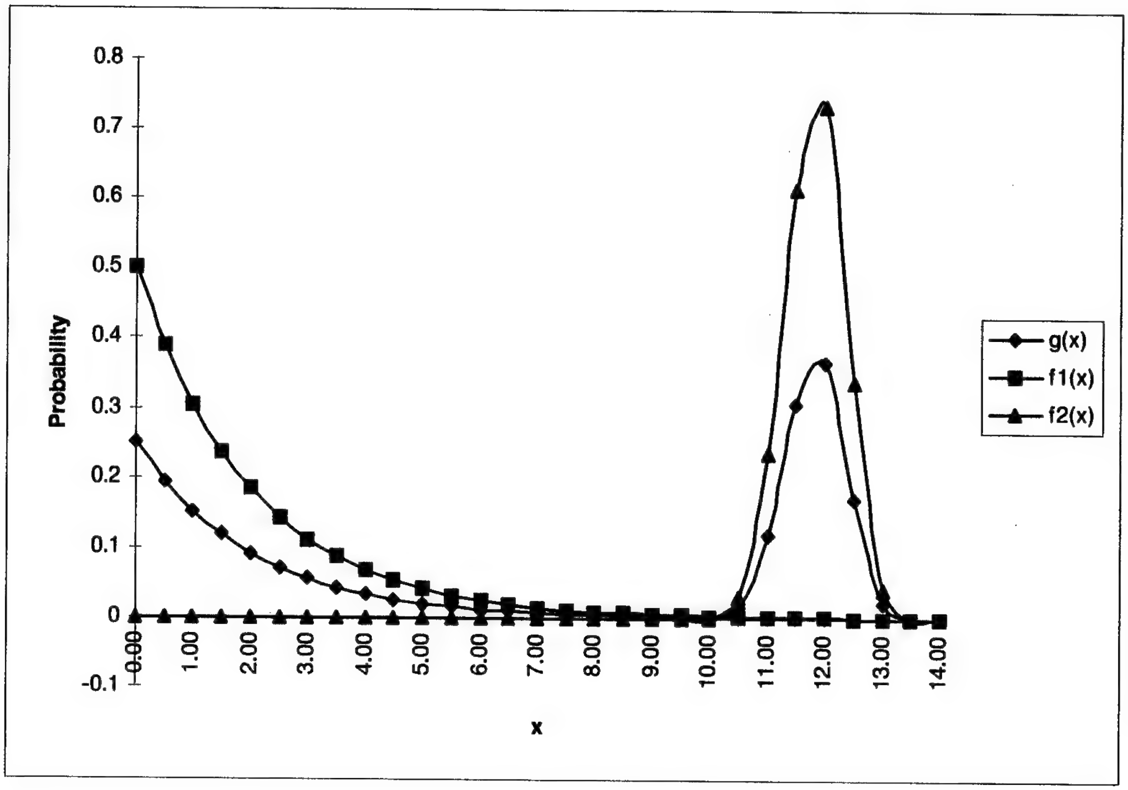

Figure 2 PDF for a Mixture of Two Generalized Gamma Distributions

The log likelihood function where g(·) is the PDF for the mixture of two Generalized Gamma Distributions and f(·) is the PDF for the single component Generalized Gamma Distribution is:

$$LL = \ln\left(\prod_{i=1}^{n} g(x_i)\right)$$

or equivalently;

$$LL = \sum_{i=1}^{n} \ln(g(x_i)) .$$

The derivatives of the log likelihood function with respect to each of the nine parameters are:

$$\frac{dLL}{dc_1} = \sum_{i=1}^{n} \frac{m}{g(x_i)} \cdot {df_1}\big/{dc_1}$$

$$\frac{dLL}{da_1} = \sum_{i=1}^{n} \frac{m}{g(x_i)} \cdot {df_1}\big/{da_1}$$

$$\frac{dLL}{db_1} = \sum_{i=1}^{n} \frac{m}{g(x_i)} \cdot {df_1}\big/{db_1}$$

$$\frac{dLL}{dp_1} = \sum_{i=1}^{n} \frac{m}{g(x_i)} \cdot {df_1}\big/{dp_1}$$

$$\frac{dLL}{dc_2} = \sum_{i=1}^{n} \frac{1-m}{g(x_i)} \cdot {df_2}\big/{dc_2}$$

$$\frac{dLL}{da_2} = \sum_{i=1}^{n} \frac{1-m}{g(x_i)} \cdot {df_2}\big/{da_2}$$

$$\frac{dLL}{db_2} = \sum_{i=1}^{n} \frac{1-m}{g(x_i)} \cdot {df_2}\big/{db_2}$$

$$\frac{dLL}{dp_2} = \sum_{i=1}^{n} \frac{1-m}{g(x_i)} \cdot {df_2}\big/{dp_2}$$

$$\frac{dLL}{dm} = \sum_{i=1}^{n} \frac{f_1(x_i,c_1,a_1,b_1,p_1) - f_2(x_i,c_2,a_2,b_2,p_2)}{m \cdot f_1(x_i,c_1,a_1,b_1,p_1) + (1-m) \cdot f_2(x_i,c_2,a_2,b_2,p_2)}$$

Table 3 First Derivatives of the GGD9

See Table 1 for the partial derivatives of single component with respect to its four parameters.

Research has also been done on mixing with the Generalized Gamma Distribution. In 1989, Chukwu and Gupta developed a discrete mixture using the Generalized Poisson and the three-parameter Generalized Gamma Distribution (7:319). In 1992, Radhakrishna, Rao and Anjaneyulu developed parameter estimates for the mixture of two three-parameter Generalized Gamma Distributions using moment estimates and Maximum Likelihood Estimates. (43:1799). They recommended using a method

such as Newton-Raphson to solve their Maximum Likelihood equations. They did not attempt to solve the equations, so they did not discuss any of the numerical problems that would arise from such an endeavor. It is believed that this work is the first to attempt to solve the Maximum Likelihood equations for the Mixed Generalized Gamma Distribution. Instead of using Newton-Raphson, this work used a Genetic Algorithm to maximize the Log Likelihood equation.

2.2 Genetic Algorithms

2.2.1 Introduction

Genetic Algorithms are a class of heuristic optimizing techniques that were developed in the 1970's by John Holland of the University of Michigan (59). They are a technique for finding a near optimal solution of an equation or system of equations by mirroring genetic theories of reproduction. At each generation, a fixed number of individuals exist, the "fittest" tend to survive and they reproduce better individuals through crossover. Crossover is process by which parts of two chromosomes are joined, hopefully improving the fitness of resulting generation. Mutations, which are random changes to the individual solution, keep a diverse population and prevent convergence from occurring too quickly. Mutations and crossover are operations that occur on a chromosome. In this optimization technique, the chromosomes represent candidate solutions to the objective function. An overview of the steps in a Genetic Algorithm are given below.

The Genetic Algorithm

1. Initialize a population of "chromosomes", or possible solutions.
2. Evaluate each chromosome in the population.
3. Create new chromosomes by mating current chromosomes, applying mutation and recombination as the parent chromosomes mate.
4. Delete members of the population to make room for the new chromosomes.

5. Evaluate the new chromosomes and insert them into the population.
6. If time is up, stop and return the best chromosome; if not, go to 3 (8:5).

The most basic Genetic Algorithm contains three operations on the chromosomes: Roulette wheel selection, simple crossover, and simple mutation. The equation to be optimized or a transformation of it is called the Fitness function. The equation must always return a positive (non-zero) value, so it may be transformed to ensure this is the case. The higher the value the fitness function has, the more "fit" the solution is. This is critical to selection of individuals for the next generation.

Holland stated reproduction of a new generation uses the following three steps:

1. Reproduction according to fitness. Select strings from the current population to act as parents. The more fit the string, the more likely it is to be chosen as a parent. A given string of high fitness may be a parent several times over.
2. Recombination. The parent strings are paired, crossed over, and mutated to produced offspring strings.
3. Replacement. The offspring strings replace randomly chosen strings in the current population. The cycle is repeated over and over to produce a succession of generations (24:70).

2.2.2 Coding

Coding is the process of converting a solution into a chromosome that genetic operations can be operated on. The coding of a solution to the equation is the conversion of the decimal base solution into a base 2 solution. For an integer variable ranging from 0 to 15, four bits would be necessary. For example, in base 10, "0" is converted to "0000" in base 2, "6" is converted to "0110", and "15" is converted to "1111". For more than one variable, a technique called mapping is used (15:82). In this case, the bit positions represent different variables. For example, for integers x and y ranging from 0 to 15, a total of eight bits are used and stacked next to one another. One particular solution x=6, y =15 would with x mapped first be "01101111". The solution x=15, y=6 under this mapping scheme would be coded "11110110". For fractional representations of variables the process is similar. For example, let the base be ¼ , then the number 2½ is "1010" where the bit positions represent 2 , 1, ½, ¼.

2.2.3 Roulette Wheel Selection

Roulette wheel selection is a method that weights the fittest individuals, so that they are more likely to be chosen for the next generation. The sum of the fitness of all individuals is calculated. The probability of selecting an individual is calculated by dividing its fitness by the sum of the fitness scores. Consider a population of three with fitness scores of 5, 8, and 7. The probability of selection is in Table 4.

Table 4 Example of Roulette Wheel Selection

Individual	Fitness	Probability	Selection Range
1	5	5/20 (0.25)	0.00-0.25
2	8	8/20 (0.40)	0.26-0.65
3	7	7/20 (0.35)	0.66-1.00
Sum	20	20/20	

The need for all fitness values to be positive is now apparent. If any fitness value is 0 or negative, it would cause some individuals to never be selected, because it would have no probability associated with it.

The individual is selected by drawing a uniform random number between 0 and 1 and comparing it to the selection range. For example, if the random number returns a 0.553, that falls in the selection range for Individual 2, so Individual 2 is returned. This is repeated until the total number of individuals for the next generation is selected.

2.2.4 Simple Crossover

Simple crossover is an operation that modifies the chromosomes of the two children that are created from the two parents chromosomes. A random integer between 1 and n-1, where n is the number of bits in the chromosome, is selected as the site where the crossover will take place (12:62-65). For example, suppose the crossover site is "3" at the "|" below for the following:

Parent Solution (x,y)	Parent Chromosome	Child Chromosome	Child Solution (x,y)
(6,15)	"011|01111"	011**10110**	(7,6)
(15,6)	**"111|10110"**	**111**01111	(14,15)

The probability of crossover is typically between 0.6 and 0.7 (5). When crossover does not occur, the children inherit the exact chromosomes of the parents, unless they are mutated. In the example above, without crossover the children's chromosomes will be "01101111" and "11110110".

2.2.5 Simple Mutation

Simple mutation is the last operator on the chromosome. When it occurs, it modifies the value at a bit position, which is also called an allele (15:65). For each allele, a random number is drawn. If the number is below the mutation probability, a mutation occurs. When this happens an allele value will be swapped. It may occur to a child's chromosome whether or not it has crossed over from its parents. Several examples with alleles to be mutated in **bold** follow:

Solution (x,y)	Chromosome	Mutated Chromosome	Mutated Solution(x,y)
(6,15)	"01101**1**11"	"01101**0**11"	(6,11)
(7,6)	01**110**110	"01**011**110"	(5,14)
(15,6)	"1111011**0**"	"1111011**1**"	(15,7)

Mutations as can be seen above, can make either small or quite large changes in the solution based on where the mutation occurs. The purpose of mutations is to prevent the population from becoming too homogeneous too quickly and thus not considering some potentially good solutions, i.e. it prevents the solution technique from converging to a local maximum.

2.2.6 Deterministic Tournament Selection Strategy

Another method of selecting individuals is the Deterministic Tournament Selection strategy. It is particularly useful when the population is small and the law of averages doesn't hold (29:290). First, individuals are grouped randomly and then adjacent pairs compete for selection. Two copies of the same individuals mating with each other should be avoided (29:291). The process is

1. Randomly sort the individuals.

2. Compare the fitness values of the first pair, and keep the better as the first parent.

3. Compare the fitness values of the second pair and keep the better of the second pair as the second parent.

4. Then randomly reorder the individuals and select parents for mating until the next generation is filled.

For example, assume the population on left of Table 5 has been randomly ordered. Individuals 1 and 2 would be compared and Individual 2 is selected as the first parent, since it has the higher value. Individuals 3 and 4 would be compared and Individual 3 is the second parent. Individuals 2 and 3 have been chosen to mate with each other. Similarly, after reordering, in the population on the right Individuals 3 and 4 will be selected.

Table 5 Example of Deterministic Tournament Selection

Individual	Fitness		Individual	Fitness
1	4		3	8
2	9	Randomly Order	5	3
3	8		1	4
4	6		4	6
5	3		2	9

2.2.7 Parameter Settings

Genetic Algorithm searches are highly sensitive to their parameter settings. Their settings greatly affect how well a GA performs. Performance measures include how long it takes to find a solution and how good a solution can be found. Greffenstette studied optimizing the control parameters of Genetic Algorithms (17:5-11). Although there are a large class of GA's, many can be described using five parameters, as shown in Table 6. The discussion in the following section will present a summary of his parameter descriptions.

Table 6 Parameter Codes

Parameter Code	Parameter
N	Population Size
C	Crossover Rate
M	Mutation Rate
G	Generation Gap
S	Selection Strategy

The population size (N) affects both the efficiency and the performance of the GA. A large population is likely to represent more areas of the sample space and thus is less likely to converge to a

suboptimal solution. However, it does require more evaluations per generation, which can drastically slow down convergence.

The crossover rate (C) controls the frequency that the crossover operator is applied. The higher the crossover rate, the more frequently new individuals are introduced into the population. If the crossover rate is too high, then high-performance individuals can be discarded. On the other hand, if the crossover rate is too low, the search may stagnate because of the lower exploration rate. It is a probability operator, so it ranges from 0 to 1.

The mutation operator (M) is a secondary search operator that gives each bit a chance of switching its value after selection. The mutation rate increases the variability of the population. A high mutation rate is essentially a random search. A low mutation rate serves to prevent any given bit position from remaining converged in the entire population. It is a probability operator, so it ranges from 0 to 1.

The generation window (G) controls the percentage of the population to be replaced at each generation. It ranges from 0 to 1. The number of individuals that are randomly chosen to survive is N*(1-G) from one generation to the next. A value of G = 1.0 means that the entire generation is replaced during each generation.

The Selection Strategy (S) contains two possibilities. The first, when S=P, is when a pure selection strategy is used. This means that each individual is reproduced proportionally to the individual's performance. The second, when S=E, is called an elitist strategy. First, the pure selection strategy is performed, but then the best individual always survives intact to the next generation. Without this strategy, the best individual can disappear because of selection, crossover or mutation.

2.2.8 Micro-Genetic Algorithms

The usual choice in population size N is usually chosen to be between 30 and 200 individuals. Micro-Genetic Algorithms (mGA) use a small sample size, far less than the typical simple Genetic Algorithm. GA's with very small populations generally do very poorly because they have insufficient processing of the many possible solutions and thus converge to suboptimal points (29:290). In Grefenstette's study, he showed that for small populations (20 to 40 individuals), good performance can be

obtained by combining a high crossover rate with a low mutation rate (17:10). Krishnakumar took this idea even further and developed a mGA that has a population N=5. He used a high crossover rate of C=1 and a low mutation rate of M=0 and an elitist selection strategy of S=E and a generation window of G=1 (29:291).

His work was based on the fact that simple Genetic Algorithms often prematurely converge and must rely on the mutation operator to find the optimum. It is based on the assumption that mixing the maximum possible solution spaces yields maximum performance. His mGA uses a "start and restart" procedure that avoids premature convergence by replacing individuals with new randomly generated ones once the individuals reach a convergence criteria (29:291).

2.2.9 Constrained Optimization Using Genetic Algorithms

Genetic Algorithms are often used to solve an unconstrained objective function. For a constrained problem, this function can be transformed to an unconstrained by use of a penalty function (15:85-86). An example follows:

Maximize g(x)

subject to $h_i(x)=0$ for i=1,2..n

where x can be a vector

into the following unconstrained form:

$$\max_x \left\{ g(x) - r \sum_i \Phi(h_i(x)) \right\}$$

where $\Phi(\cdot)$ is the penalty function

r is the penalty coefficient.

Many possibilities exist for the penalty function, but often the square of the violation is used. The value for r is often chosen so that moderate violations yield a penalty of some significance (15:85-86).

2.2.9 Initializing a Population

Previous discussion has focused on the changes from generation to generation, whether through mutation, crossover, or "restarting". An initial population can be chosen by random or by a heuristic (17:5). In random initialization, possible solutions are randomly generated with equal probabilty for each allele on every generated chromosome. For example, with two possible allele settings "0" or "1" each has a 0.5 probability of being selected. A heuristic would mean that one or more individuals are created with some other rule, not specific to the Genetic Algorithm.

2.3 Maximum Likelihood Estimation

The Method of Maximum Likelihood was popularized by Fisher in the 1920's (38:2-3). The definition of likelihood from Mendenhall, Wackerly and Scheaffer (36:402) is

> Let $y_1, y_2, \ldots, y_n$ be sample observations taken on corresponding random variables Y_1, $Y_2, \ldots Y_n$. Then if $Y_1, Y_2, \ldots, Y_n$ are continuous random variables the likelihood $L=L(y_1, y_2, \ldots y_n)$ is defined to be the joint density evaluated at $y_1, y_2, \ldots, y_n$.

Parameters are selected so the likelihood function is maximized (36:419). This means that the likelihood function is defined as follows:

$$L = f(x_1) * f(x_2) * \cdots * f(x_n)$$

or alternatively,

$$L = \prod_{i=1}^{n} f(x_i)$$

where $f(x_i)$ above is the probability density function associated with the observation x_i and n is the sample size.

According to Law and Kelton, the Maximum-Likelihood estimators (MLEs) are used because they have useful properties not shared by other parameter estimation techniques (45:370). These include:

1. For most of the common distributions, the MLE is unique; that is, L($\hat{\theta}$) is strictly greater than L(θ) for any other value of θ.

2. Although MLEs need not be unbiased, in general, the asymptotic distribution (as n $\rightarrow \infty$) of $\hat{\theta}$ has mean equal to θ.

3. MLEs are invariant; that is if $\phi = h(\theta)$ for some function h, then the MLE of ϕ is h($\hat{\theta}$). (Unbiasedness is not invariant.) For example, the variance of an Exponential(B) random variable is B^2, so the MLE of this variance is $[\overline{X}(n)]^2$.

4. MLEs are asymptotically normally distributed; that is, $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{D} N(0, \partial(\theta))$, where $\partial(\theta) = -n / E(d^2 L / d\theta^2)$ (the expectation is with respect to X_i, assuming that X_i has the hypothesized distribution) and $\xrightarrow{D}$ denotes convergence in distribution. Furthermore, if $\hat{\theta}$ is any other estimator such that $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{D} N(0, \sigma^2)$, then $\partial(\theta) \leq \sigma^2$. (Thus, MLEs are called best asymptotically normal.)

5. MLEs are strongly consistent; that is, $\mathrm{Lim}_{n \rightarrow \infty} \hat{\theta} = \theta (\mathrm{w.p.1})$.

L is the Likelihood equation and θ is the vector of parameters.

2.4 Minimum Distance Estimation

In the 1950's, Wolfowitz presented a series of papers that developed the Minimum Distance method for obtaining strongly consistent parameter estimates for a distribution (25:75). His technique was to minimize the distance of the discrepancy between two distributions F_1 and F_2:

$$\delta(F_1, F_2) = \sup_x |F_1(x) - F_2(x)|$$

According to Wolfowitz, "A great utility of the Minimum Distance method is that, in a wide variety of problems, it will furnish super-consistent estimators even when classical methods, like maximum likelihood

method, fail to give consistent estimators" (38:2-7). The technique tries to minimize the distance between the estimated cumulative density function (CDF) and the empirical density function (EDF). The EDF is a step function created by ordering the sample data points. One plotting position for the EDF of n points ordered $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$ is given by

$$EDF(X) = \begin{cases} 0, & x < x_{(1)} \\ \frac{i}{n}, x_{(i)} \le x < x_{(i+1)}, & i = 1, \ldots, (n-1) \\ 1, & x \ge x_{(n)} \end{cases}$$

A good estimate of the parameters for the estimated cumulative density function must be derived from another method, such as MLE, and the better the method's estimate the better result that Minimum Distance can give. A good initial estimate of the parameters may be obtained using the Method of Maximum Likelihood. An example of the EDF compared to the CDF is Figure 3.

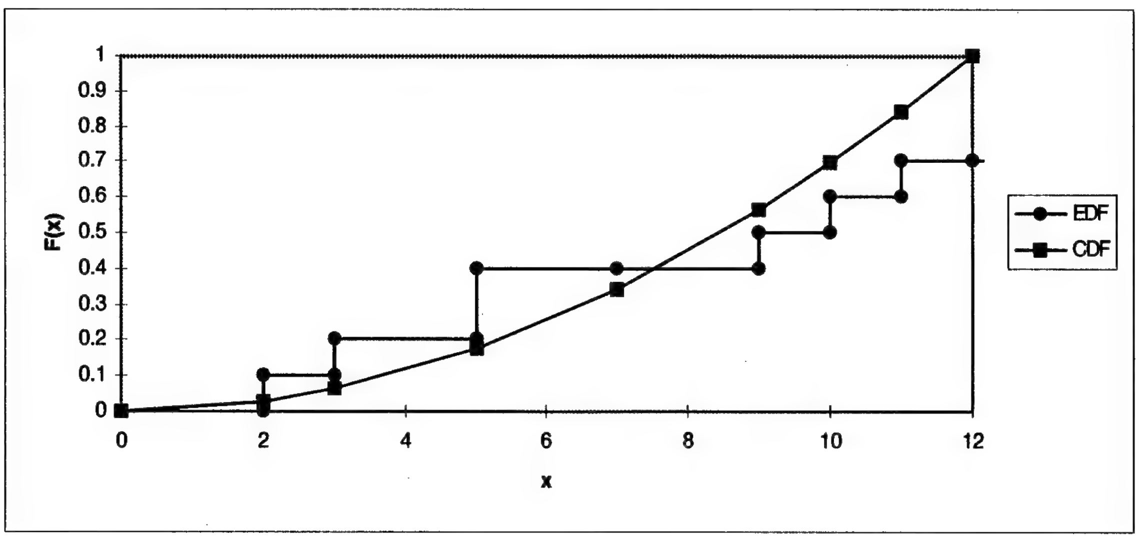

Figure 3 Sample EDF vs. Estimated CDF

A lot of work has been done with Minimum Distance, with some of it already referenced in this thesis. In 1980, Hobbs, Moore and James used Minimum Distance to estimate the parameters for the three-parameter Generalized Gamma Distribution (26:vi, 23:237). In 1982, Shumaker used Minimum Distance to

estimate the parameters for the four-parameter Generalized Gamma Distribution (49:22). Gallagher used Minimum Distance to estimate the parameters for the three-parameter Weibull (14). In 1986, Benton-Santo used Minimum Distance to estimate the parameters for the mixture of exponential distributions and the mixture of normal distributions (1). In 1996, Mumford used Minimum Distance to estimate the parameters for the seven-parameter Mixed Weibull(37).

2.4.1 Golden Section Search

One method of performing Minimum Distance is to vary one parameter between its upper and lower bounds, while fixing the others, to minimize the distance between the EDF and Estimated CDF. One method of conducting a constrained optimization of a single variable is called the Golden Section Search. It assumes that the function being evaluated is unimodal in the area of search. It uses an interval reduction factor based on the Fibonacci numbers (48:115-116,122-123;30:286-291). The algorithm is given by (48:115-116):

Given:

- The interval [A,B], which contains the minimum value for function f(x)
- The tolerance level, Tol
- The maximum number of iterations, N

Algorithm:

1. Set Iter = 0
2. Set $t = \dfrac{\sqrt{5}-1}{2}$
3. Set C = A+ (1-t) * (B-A)
4. Set Fc = f(C)
5. Set D = B - (1-t) * (B-A)
6. Set Fd = f(D)
7. Repeat the next steps until |B-A| > Tol and Iter < N;

7.1. Increment Iter

7.2. If Fc < Fd, then set B=D, D=C, C=A+(1-t)*(B-A), Fc = f(C); else set A=C, C=D,D=B-(1-t)*(B-A), Fc=Fd, and Fd=f(D).

8. Return $\frac{A+B}{2}$ as the minimum if Iter < N, else return an error code.

2.5 Random Variate Generation

Random variate generation for the Generalized Gamma distribution is accomplished by generating random variates from a standard gamma distribution, which are then transformed to Generalized Gamma distribution variates. Numerous techniques exist for generating gamma and standard gamma variates (6, 13, 27, 28, 33, 55, 56, 57, 59, 58, 62, 63). The probability density function for the standardized gamma distribution is

$$f(x;a) = \frac{x^{a-1} \cdot \exp(-x)}{\Gamma(a)}, \quad x \geq 0$$

Four-parameter Generalized Gamma variates can be generated from a standard gamma using two transforms demonstrated by Tadikamalla (58: 199-201). The transforms are

$$x = z^{1/p}$$

and

$$y = x * a - c$$

where z is the variate generated from the standardized gamma distribution and y is the final Generalized Gamma Distribution variate. The standardized gamma distribution generates its variates using the Acceptance/Rejection technique.

The Acceptance/Rejection technique requires a majorizing function, h(t) which bounds f(t) above. The majorizing equation must integrate to a finite value so that it may be scaled as a PDF. Variates are then

generated from h(t) and are then accepted or rejected so that the accepted random variates will have the PDF f(t) (33:83).

III. Methodology

This research compares two methods for estimating the parameters of the nine-parameter Mixed Generalized Gamma Distribution: Maximum Likelihood Estimation and Minimum Distance Estimation. The assumption underlying this methodology is that the distribution is from the family of nine-parameter Mixed Generalized Gamma Distributions. Monte Carlo analysis will be used to test the parameter estimation techniques. Random variates are generated from the Mixed Generalized Gamma Distribution, and then the parameters are estimated using Maximum Likelihood. The log likelihood function was maximized using a Genetic Algorithm. The Maximum Likelihood parameter estimates are then used as a initial estimate for the Minimum Distance parameter estimates. This initial estimate is then used to fix the first location parameter and the mixture parameter. This estimation technique fixes the first location and mixture parameter and solves the reduced problem using Maximum Likelihood. Samples of each size and parameter settings were generated 1000 times and then compared using a integrated mean square error (MSE) , or integrated squared distance between CDFs. The number of times one technique was better than the other technique for estimating each sample was recorded. This was done to guard against a few extreme cases from dominating the comparison of the two parameter estimation techniques.

3.1 Monte Carlo Simulation

Law and Kelton define Monte Carlo simulation to be a scheme employing random numbers for solving problems in which the passage of time plays no substantive role (31:113). Monte Carlo simulation has been widely used to study the properties of robust estimators and to test their performance (26:19). By using Monte Carlo simulation, parameter estimates may be calculated and then compared to the true parameters of the underlying distribution. The basic steps in a Monte Carlo simulation for testing parameter estimation techniques are as follow:

1. Generate sample variates from the selected underlying distribution.

2. Determine the parameter estimates for each sample, using each estimation technique.

3. Compare the performance of the estimators.

Figure 4 shows an overview of the methodology.

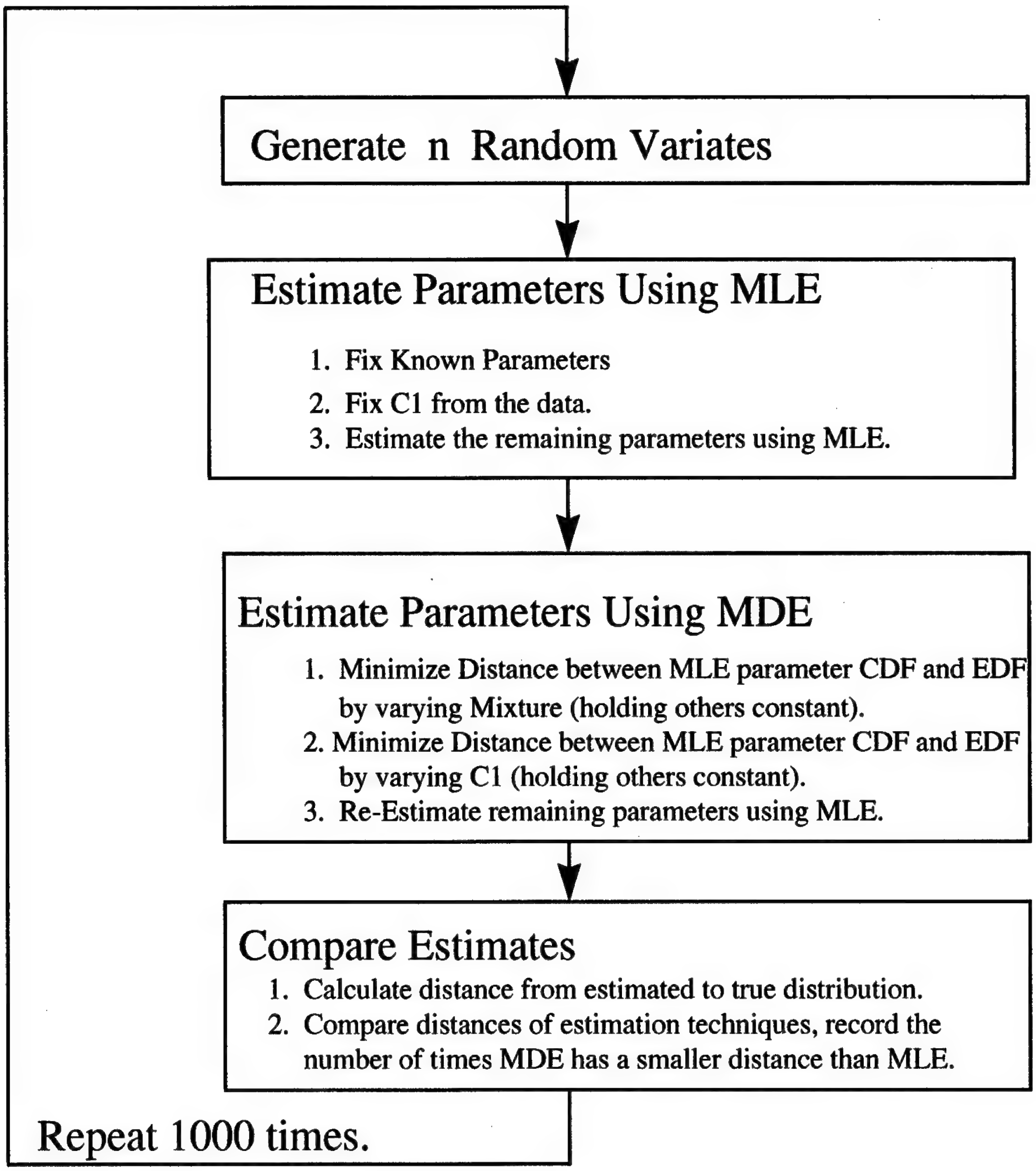

Figure 4 Overview of Methodology

3.2 Random Variate Generation

The first random variate generator for the Generalized Gamma distribution tested was one developed by Tadikamalla, who had written several articles on generating gamma variates (55-58). His random variate generator did not produce the proper variates. It did not generate variates of the proper shape when the shape/power parameter, b, was set to one. Setting b=1defines a Weibull distribution, which is an important special case of the Generalized Gamma, particularly for reliability modeling. Although Tadikamalla's article stated it was good for b>1, it was not good for values near one (58:200). Further research led to another standard gamma variate generator that was then transformed to the General Gamma distribution using Tadikamalla's transformations above.

The standard gamma variate generator chosen is known as GBH, which uses the acceptance/rejection technique, and was developed by Cheng & Feast (59:229). Tadikamalla and Johnson recommended this technique when a large number of variates are going to be generated for each b (59:226). It was also chosen because variates with b = 1 can be generated. The random variate generator was then constructed and tested by generating samples of 1000 variates of selected special cases such as the Weibull, the Gamma, the Exponential and the Half-Normal. (See Table 1). Accepted Distribution fitting packages such as Weibull++ and BestFit were used to verify the generator was working properly.

In the real world, the proportions in a mixture of component distributions are not always known. The mixing proportion is distributed Uniform(0,1). The outcome of this random number draw determines the number of variates generated from each component distribution. This means that the actual proportion of variates generated from each component distribution may not equal the true mixing proportion.

3.3 Maximum Likelihood Estimation using a Genetic Algorithm

Use of a Genetic Algorithm was suggested because of the difficulties Mumford (37) had solving the maximum likelihood equations. His seven-parameter estimation solution space turned out to be flat with many local optima. The nine parameter Mixed Generalized Gamma Distribution would have these difficulties compounded since two more parameters are being estimated. Newton's Method, which Mumford used, also requires derivative information, and can get stuck on local maximums. It was decided

to attempt to find the optimum using a Genetic Algorithm, which is less likely to get caught on a local maximum.

Since the observed data points are independent of each other, their joint probability can be calculated directly by multiplying their PDFs together. Finding the parameters that maximize their joint probability will give the parameters that have the highest likelihood that the data came from that distribution. Thus, the equation is called the likelihood function. The likelihood function (L) is defined below

$$L = \prod_{i=1}^{n} f(x_i),$$

where f(·) is the probability density function and n is the number of observed data points. This equation, though, can be difficult to differentiate and can also create numerical difficulties because it can evaluate to very small numbers that can be below a computer's "underflow." Therefore, the log of the likelihood function, which monotonically increases with the likelihood function, poses no such numerical difficulties, and is often easily differentiable, is used. The log likelihood (LL) is defined as

$$\mathrm{LL} = \ln(L) = \ln(\prod_{i=1}^{n} f(x_i))$$

and is also equivalent to

$$\ln(L) = \sum_{i=1}^{n} \ln(f(x_i)),$$

where f(.) is the probability density function and n is the number of observed data points.

3.3.1 Range of the Parameters

In order to use a Genetic Algorithm, upper and lower bounds on the variables being optimized must be determined. An increment size of 0.015625 was used for each of the parameters. See Table 7 for the parameter bounds.

Table 7 Parameter Bounds

Parameter	*Lower Bound*	*Upper Bound*
c_1	0	$X_{(1)}$ - 0.0078125
a_1	0.0078125	15.9921875
b_1	0.0078125	15.9921875
p_1	0.0078125	15.9921875
c_2	c_1 + 0.0078125	c_1+15.9921875
a_2	0.0078125	15.9921875
b_2	0.0078125	15.9921875
p_2	0.0078125	15.9921875
m	0.0078125	0.9921875

$X_{(1)}$ is the first order statistic of the variates, or the smallest variate, and $X_{(n)}$ is the last order statistic, or the largest variate. The c_2 parameter was penalized if it exceeded the largest variate since if $c_2 > X_{(n)}$ means the GA was fitting the observed data as a single distribution, not as a mixture. The mixture parameter m will need six bits on the chromosome. The bit positions will represent ½, ¼, 1/8, 1/16, 1/32 and 1/64. The bits are decoded and then the lower bound is added. For example, the upper limit of the mixing parameter is coded "111111" and will equal 0.9921875. The location parameter c_1 will not be estimated by the Genetic Algorithm, but will be calculated as Max ($X_{(1)}$ - 0.0078125, 0). The other parameters will require 10 bits each on the chromosome, when not fixed prior to estimation. The ten bit positions will represent 8,4, 2,1, ½, ¼, 1/8, 1/16, 1/32 and 1/64. The chromosome is decoded and the lower bound is added. For example, the upper limit on a_1 is coded "1111111111" and will equal 15.9921875.

3.3.2 Selecting GA Parameter Settings

Selection of the Genetic Algorithm's settings greatly affect its efficiency and accuracy (17). The optimal settings are often problem dependent. Two characteristics of parameter estimation for the Mixed Generalized Gamma Distribution log likelihood function discussed below are that it will be a negative number and that it will be complex to evaluate.

First, the Mixed Generalized Gamma distribution log likelihood equation will be negative. Since probability density function will typically evaluate to a number between zero and one. Multiplying n of these PDFs together returns a value between zero and one. The natural logarithm of any number between

zero and one will have a negative natural logarithm. Thus, any solution technique used must work with negative numbers. However, the simple tournament selection requires that all values be positive. This can be resolved by transforming the negative numbers using a scaling window, though choosing the right scaling is problematic since the solution can be very sensitive to the scaling (17:7). As an example, if one individual has a fitness five times greater than another, it would have an expected five times as many offspring. Scaling can modify this in ways that are not effective. For example, consider the following scaling: y=2*(x+20) in Table 8

Table 8 Dangers of Scaling

Individual	Fitness	Relative Fitness	Scaled Fitness	Related Scaled Fitness
Better	-1	5	38	1.26
Worse	-5	1	30	1

The Better Individual, which was previously five times more fit, is now only 1.26 times as fit. The less fit individual will reproduce relatively more with respect to the better individual as a result of the scaling. This can even eliminate the difference between fit and less-fit strings (15). One way to avoid this problem is to use a deterministic tournament selection strategy.

A Micro-Genetic Algorithm was chosen to maximize the log likelihood function of the Mixed Generalized Gamma Distribution. It typically uses fewer function evaluations than a regular Genetic Algorithm, and it can easily use a deterministic tournament selection strategy so the need to scale is eliminated. A deterministic tournament selection strategy compares the relative fitness values not the absolute fitness as the tournament selection does, so negative numbers are compared directly without any need to scale the fitness. Thus, in this work, a Micro-Genetic Algorithm was used to maximize the log likelihood equation. The population size chosen was five individuals per generation, as recommended by Krishakumar (29). One of the most referenced works on Micro-Genetic Algorithms is by Krishakumar. He states the key to making a Genetic Algorithm with a small population is as follows:

1. Randomly generate a small population.
2. Perform Genetic operations until nominal convergence (as measured by bit wise convergence or some other reasonable measure).

3. Generate a new population by transferring the best individuals of the converged population to the new population and then generating the remaining individuals randomly.

4. Go to step 2 and repeat (29:290).

The Micro-Genetic Algorithm first randomly generated a population of five. Two sets of mates were chosen to reproduce by crossover with probability one. An individual was not allowed to mate with itself, as suggested by Krishakumar (29:291). The probability of mutations was zero, since enough diversity is introduced after nominal convergence. Nominal convergence is when all the individual chromosomes are virtually identical. This was defined to be when the all the individuals in a population had at least 95% of the same allele structure (5). At this point, crossover no longer introduces new chromosome structures. Therefore, four new individuals were then introduced and the best string kept as the fifth. The stopping criteria is to check every 200th generation and then stop if the best individual has not changed after that amount of time.

3.4 Minimum Distance

Once a parameter estimate was found using Maximum Likelihood Estimation, this estimate was used to initiate the Minimum Distance parameter estimation technique. The other eight parameters were held constant and the mixing parameter was varied to find where it minimized the distance between the empirical probability function and the estimated probability distribution function. A Golden Section Search was performed to find the function minimum. A Golden Section Search locates the minimum within the interval using an interval reduction technique based on interval reduction (48:115). The lower bound on the interval was the lowest value the variable could take on and the upper bound the highest, the bounds are in Table 9.

Table 9 Parameter Bounds for Minimum Distance

Parameter	*Lower Bound*	*Upper Bound*
m	0.0078125	0.9921875
c_1	0	$X_{(1)}$ -0.0078125
c_2	c_1+0.0078125	$X_{(n)}$ - 0.0078125

The mixture parameter(m) was allowed to vary and the other parameters were held constant. The value of the mixing proportion that had the smallest Anderson-Darling statistic was kept. Next, c_1 was similarly found. These two parameters were fixed. The other parameters were again estimated using Maximum Likelihood Estimation with the mixture parameter and the first location parameter fixed at their MDE values.

The Anderson-Darling statistic is one of the most powerful empirical distribution function-based tests and takes the form:

$$A_n^2(G_n, F_\theta) = \int_{-\infty}^{\infty} [G_n(x) - F_\theta(x)]^2 [F_\theta(x)[1 - F_\theta(x)]]^{-1} dF_\theta(x)$$

This integral is approximated using Stephens' computational formula (54:731)

$$A^2 = -\left\{\left[\sum_{i=1}^{n} (2i-1)[\ln Z_i + \ln(1 - Z_{n+1-i}) / n\right]\right\} - n$$

where $Z_i = F_\theta(x_i)$, $F_\theta(x)$ is an estimated distribution using the maximum likelihood method and G_n is an empirical distribution based upon a random sample of size n that is taken from the true distribution G($\cdot$) (38: 2-8 to 2-10).

Now, with the mixture parameter fixed, the c_1 location parameter was then fixed. The other parameters are estimated again using the Micro-Genetic Algorithm to find the final Minimum Distance estimate. Attempts to fix the c_2 location parameter only worsened parameter estimates, because Minimum Distance selected c_2 parameter values that were very high. The parameter estimates using the Method Of Maximum Likelihood and Method Of Minimum Distance were then compared to the true parameters that the samples were generated from using the evaluation criteria below.

3.5 Evaluation Criteria

Once the two parameter estimates for a sample of random variates have been calculated, they need to be compared so that the better method of the two may be determined. A good criteria is to compare how close each parameter estimate is to the true parameter distribution. A distance between the CDF of the

estimation technique and the CDF of the true parameters was calculated for both of the estimation techniques. The better technique will have the smaller distance.

3.5.1 Integrated Squared Difference Between CDF's

The estimated CDF for each method is compared to the true CDF. The integrated squared distance between the estimated CDF and the true CDF is calculated as the measure of performance. It is also known as the Integrated Mean Squared Error. This test is used because of its effectiveness and it approximated the theoretical Cramer-von Mises statistic (37:41). If the integrated difference is smaller for one method than the other, then it has been a better estimate. The integration is

$$\text{dist} = \int_a^b \{F(x)-G(x)\}^2\,dx$$

where F(x) represents the true CDF and G(x) represents the estimated CDF. The lower limit of integration, a, is the smaller of the two true location parameters. The upper limit of integration, b, is chosen to ensure that both CDF values exceed 0.999, or an upper limit of 50, which ever was smaller (37:40-41).

The numerical integration algorithm used was Gauss-Legendre Quadrature. It integrates using the transformation

$$\int_a^b f(x)dx = \tfrac{1}{2}\cdot(b-a)\int_{-1}^{+1} f\left[\tfrac{1}{2}\cdot(b-a)\cdot t+b+a\right]dt$$

This integral is approximated by evaluating f(x) at six points. Accuracy can be improved by breaking the integral in subintervals, with each subinterval still requiring six function evaluations (48:81-82,89).

The evaluation criteria is the percentage of times that the MDE was better than MLE. "Better" in this case means having a smaller integrated distance from the true parameters than the distance the MLE parameters were from the true parameters. This criteria was chosen because it is insensitive to outliers.

IV. Results

In order to evaluate the performance of the parameter estimation techniques, Monte Carlo simulation was performed. For each true parameter setting and sample size (n), a sample of random variates was generated from the Mixed Generalized Gamma Distribution. Parameter estimates using both techniques were then made and the MSE, or integrated distance, from each estimate to the true parameters was calculated. The distances were then compared and the technique with the smaller distance from true was considered the better. In case of a tie, MLE is better because MDE has not gained anything for the effort of refining the MLE estimate. One thousand Monte Carlo replications were run. The average distance (Ave Dist) and standard deviation of the distance (StDev Dist) are recorded in the following tables in this chapter. The percentage of times that the Minimum Distance Estimation technique (%MDE Better) had a smaller distance from the true parameters than the Maximum Likelihood Estimation were recorded. If this percentage is high, then the MDE technique has been shown to be a better estimator. Following each of the tables is a chart containing the Percent MDE Better for each sample size. The different bars at each sample size represent the different Percent MDE Better at each mixing proportion.

Special cases of the Mixed Generalized Gamma Distribution of progressing difficulty were tested to show the validity of the estimation techniques. First, the parameters for a single distribution with a known location parameter were estimated using MLE. Next, the parameters for a single distribution with unknown location parameters were estimated, using MLE and using MDE to fix the location parameter. Next, the parameters for two component mixtures of Exponential distributions and Weibull distributions with known location parameters were estimated using MLE and using MDE to fix the mixture parameter. Next, the parameters for two component mixtures of the Exponential and Weibull distributions with unknown location parameters were estimated using MLE and MDE. MDE was used to fix the mixture parameter and then the location parameter of the component distribution nearer to zero. Lastly, the parameters for the full Mixed Generalized Gamma distribution were estimated using MLE and MDE. MDE was used to fix the mixture parameter and then the location parameter of the component distribution nearer to zero. PDFs for the component distributions may be found in Appendix A.

4.1 Single Component Distribution with Known Location Results

Four special cases of the Generalized Gamma distribution were estimated. They were the Exponential, Gamma, a Weibull with a power parameter less than one, and a Weibull with a power parameter greater than one. Previous research in Minimum Distance by other authors showed that it has the most effect on estimating mixture parameters and location parameters. Since these are single component distributions (thus, no mixing parameter) with known location parameters, Minimum Distance estimation was not performed.

Several parameters were fixed prior to estimation. For all the distributions the location parameter was fixed, since it was known. It was assumed that the functional form of the component distribution was known. Therefore, for the Exponential, the parameters b and p of the GGD4 were fixed at 1; for the Gamma, the parameter p was fixed at 1, and for the Weibull, the parameter b was fixed at 1. Parameter estimation for the Weibull and Gamma was aided by use of a penalty function which penalized differences from the likelihood equation first derivatives as defined by Parr & Webster, which assumes a known location parameter (39).

Maximum Likelihood Estimation behaved exactly as expected for the four distributions. As sample size got larger, the average distance from the true distribution decreased in all cases. The standard deviation also decreased as sample size went up, which means that MLE techniques improves with sample size. This is as expected, since MLE is known to improve asymptotically. They were compared to sample results using Weibull++, which is an accepted package for estimating parameters. The results are summarized in Table 10.

Table 10 Single Component Distributions, with Known Location Parameters, 1000 Replications

Distribution	*n*	*Ave Dist*	*StDev Dist*	*Sample Estimate Using W++ or BestFit*	*Sample Distance*
GGD4(0,2,1,1)	5	0.0409	0.1041	Expo(2.394)	0.00369
Equivalent to:	20	0.0067	0.0115	Expo(2.055)	0.00009
Expo(2)	50	0.0024	0.0035	Expo(1.983)	0.00001
	75	0.0017	0.0029	Expo(1.914)	0.00025
	100	0.0013	0.0019	Expo(1.875)	0.00054
	500	0.0002	0.0003	Expo(1.949)	0.00008
GGD4(0,2,2,1)	5	0.0719	0.0524	Gamma(5.48,0.57)	0.18940
Equivalent to:	20	0.0286	0.0386	Gamma(2.85,1.41)	0.00605
Gamma(2,2)	50	0.0092	0.018	Gamma(2.10,1.81)	0.00171
	75	0.0066	0.0136	Gamma(2.02,1.91)	0.00059
	100	0.0043	0.0092	Gamma(2.06,2.01)	0.00018
	500	0.0008	0.0017	Gamma(2.14,1.77)	0.00220
GGD4(0,0.5,1,0.9)	5	0.4385	0.7672	Weib(0.623,0.584)	0.06943
Equivalent to:	20	0.0901	0.1032	Weib(0.762,0.521)	0.02951
Weibull (0.9, 0.5)	50	0.0344	0.0439	Weib(0.948,0.555)	0.01239
	75	0.0223	0.0273	Weib(0.861,0.541)	0.00238
	100	0.0178	0.0199	Weib(0.878,0.529)	0.00105
	500	0.0045	0.0069	Weib(0.884,0.495)	0.00068
GGD9(0,2,1,2)	5	0.1214	0.1943	Weib(2.171,2.334)	0.01532
Equivalent to:	20	0.0191	0.021	Weib(2,323,2.053)	0.00607
Weibull(2, 2)	50	0.0066	0.0075	Weib(2.036,1.818)	0.00531
	75	0.0044	0.0049	Weib(1.986,1.896)	0.00162
	100	0.0029	0.0031	Weib(1.999,2.000)	0.00000
	500	0.0006	0.0007	Weib(1.989,1.970)	0.00014

Fixed Parameters:

Exponential; C=0,B=1,P=1
Gamma; C=0, P=1
Weibull; C=0,B=1

Note that the Average distance and average standard deviation are for 1000 replications, whereas the sample using Reliasoft's Weibull++ 5.0 or BestFit 1.0 is a single sample.

BestFit was used to estimate the Gamma Distributions; the other using Weibull++.

4.2 Single Component Distribution with Unknown Location Results

Again, four special cases of the Generalized Gamma distribution were estimated. They were the Exponential, Gamma, a Weibull with a power parameter less than one, and a Weibull with a power parameter greater than one. Minimum Distance Estimation was performed on the location parameter.

It was assumed that the functional form of the component distribution was known. Therefore, for the Exponential, the parameters b and p were fixed at 1; for the Gamma, the parameter p was fixed at 1; and for the Weibull, the parameter b was fixed at 1. The location parameter was fixed using Minimum Distance estimation. The remaining parameters were re-estimated using MLE. Parameter estimation for the Weibull and Gamma was aided by use of a penalty function which penalized differences from the first derivatives of the likelihood equation as calculated by Parr & Webster, which assumes a known location parameter (39).

Both estimation techniques improved their distance with increased sample size. MDE uses MLE as an initial estimator, so this is expected. Minimum Distance should show the most gain at smaller sample sizes, because MLE improves with increasing sample sizes. For the Exponential, MDE was significantly better than MLE for sample sizes of 5, but not for any higher sample sizes. For the Gamma, MDE was significantly better than MLE for all sample sizes. For the Weibull with shape parameter less than one, MDE provided significant improvements for sample sizes of 50 or less. For the Weibull with shape parameter greater than one, MDE provided significant improvements for sample sizes up to 100. The results are summarized in Table 11 and Figure 5.

Table 11 Single Component Distributions with Unknown Location Parameter, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD4(5,2,1,1)	5	68.6%	0.1144	0.2159	0.1007	0.1899
Equivalent to:	20	49.2%	0.0096	0.0151	0.0160	0.0180
Expo(2,5)	50	49.4%	0.0029	0.0043	0.0076	0.0096
	75	47.9%	0.0017	0.0027	0.0052	0.0079
	100	45.6%	0.0014	0.0023	0.0048	0.0075
	500	44.0%	0.0003	0.0004	0.0009	0.0034
GGD9(5,2,2,1)	5	100.0%	0.4039	0.3398	0.1975	0.2284
Equivalent to:	20	94.6%	0.0274	0.0270	0.0186	0.0210
Gamma(2,2,5)	50	89.3%	0.0076	0.0070	0.0064	0.0061
	75	88.4%	0.0048	0.0041	0.0044	0.0038
	100	81.3%	0.0035	0.0028	0.0033	0.0027
	500	66.7%	0.0009	0.0007	0.0009	0.0007
GGD4(5,2,1,0.9)	5	96.3%	0.2484	0.2209	0.1730	0.2076
Equivalent to:	20	67.2%	0.0275	0.0356	0.0249	0.0312
Weibull(0.9,2,5)	50	55.7%	0.0089	0.0123	0.0096	0.0124
	75	49.9%	0.0056	0.0070	0.0064	0.0076
	100	48.4%	0.0040	0.0049	0.0046	0.0057
	500	23.6%	0.0012	0.0015	0.0015	0.0028
GGD4(5,2,1,2)	5	100.0%	0.4437	0.3622	0.2711	0.3229
Equivalent to:	20	86.8%	0.0409	0.0408	0.0307	0.0336
Weibull(2,2,5)	50	83.5%	0.0120	0.0118	0.0106	0.0106
	75	76.3%	0.0074	0.0069	0.0069	0.0064
	100	60.4%	0.0056	0.0051	0.0054	0.0048
	500	49.3%	0.0010	0.0010	0.0010	0.0010

Fixed Parameters:

Exponential; B1=1,P1=1
Gamma; P1=1
Weibull; B1=1

Parameters fixed by MDE:

Location, C1, for all distributions

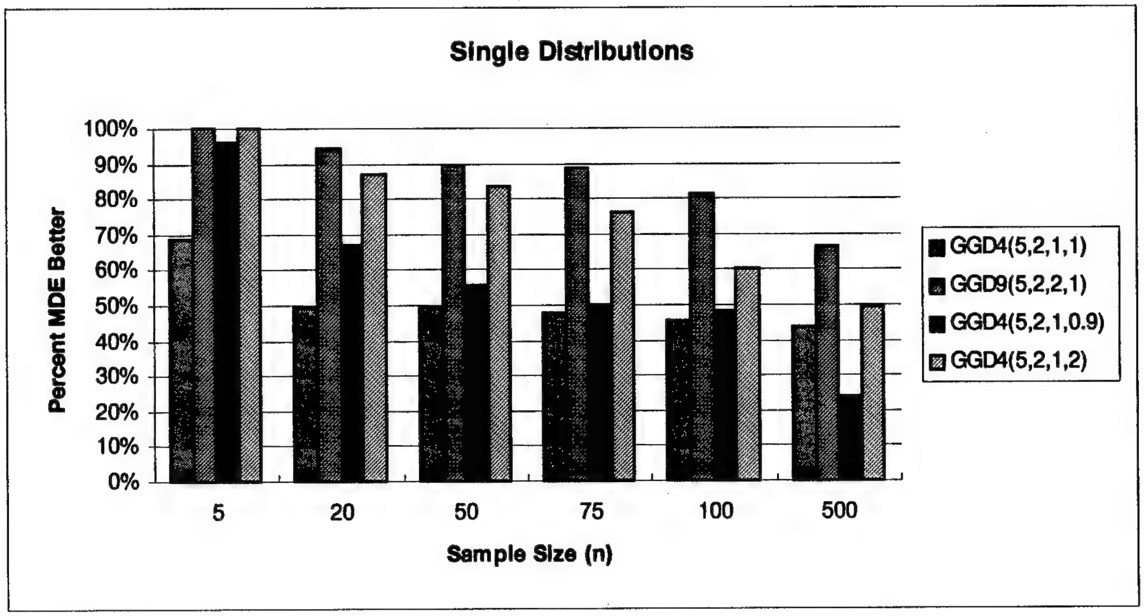

Figure 5 Percent MDE Better for Single Distributions

4.3 Mixture of Exponential Distributions with Known Location Results

For the case of the mixture of Exponential Distributions with known location parameters, it was assumed that the functional form of the distribution was known. Therefore, B1,P1, B2, P2 were all fixed at one. Since the location parameters were known, they were fixed at their known values of zero. The mixture parameter was fixed using the Minimum Distance Estimation.

For all tested mixtures, MDE was significantly better than MLE for samples of 5. For sample sizes larger than that, however, the results are unclear. Average distance and standard deviation for both parameter estimation techniques decrease as sample size increases. Typically, the larger the sample size, the less often MDE will beat MLE, but that is not the case here. In fact, Minimum Distance increased its percentage of wins from sample sizes of 100 to sample sizes of 500. These results are summarized in Table 12 and Figure 6.

Sample sizes of 750 were also made to compare with previous research using Minimum Distance by Benton-Santo for parameter estimation of the mixture of Exponential distributions (1:33). The Maximum Likelihood Estimators used in this thesis provided better estimates than the Method of Moments that she used. Her results showed that Minimum Distance did not help improve parameter estimation. The results in Table 13 and Figure 7 show slight improvement by Minimum Distance over Maximum Likelihood. The reason is that Maximum Likelihood provides a better estimate to begin with to initialize the Method of Minimum Distance.

Table 12 Mixture of Exponential Distributions, Known Location Parameters, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD9(0,0.5,1,1, 0,2,1,1, 0.25)	5	61.1%	0.1496	0.2814	0.1254	0.2437
Equivalent to	20	54.9%	0.0361	0.0593	0.0344	0.0581
0.25 Expo(0.5) + 0.75 Expo(2)	50	53.6%	0.0157	0.0199	0.0153	0.0187
	75	48.1%	0.0106	0.0137	0.0109	0.0141
	100	48.2%	0.0086	0.0115	0.0090	0.0128
	500	52.2%	0.0022	0.0026	0.0021	0.0026
GGD9(0,0.5,1,1, 0,2,1,1, 0.5)	5	57.7%	0.2399	0.5117	0.2141	0.4859
Equivalent to	20	51.8%	0.0539	0.0894	0.0518	0.0839
0.50 Expo(0.5) + 0.50 Expo(2)	50	50.2%	0.0191	0.0274	0.0197	0.0306
	75	53.3%	0.0126	0.0154	0.0127	0.0158
	100	52.4%	0.0109	0.0145	0.0109	0.0147
	500	56.0%	0.0022	0.0027	0.0021	0.0025
GGD9(0,0.5,1,1, 0,2,1,1, 0.75)	5	55.1%	0.2799	0.5469	0.2526	0.4957
Equivalent to	20	47.9%	0.0617	0.1140	0.0617	0.1141
0.75 Expo(0.5) + 0.25 Expo(2)	50	50.5%	0.0211	0.0364	0.0216	0.0381
	75	54.3%	0.0150	0.0253	0.0148	0.0240
	100	51.5%	0.0098	0.0147	0.0095	0.0138
	500	53.5%	0.0020	0.0027	0.0019	0.0024

Fixed Parameters:

Mixed Exponential; C1=0, B1=1, P1=1, C2=0, B2=1, P1=1

Parameters fixed by MDE:

Mixed Exponential; M

Table 13 Mixture of Exponential Distributions, 750 Variates, 1000 Replications

Distribution	n	%MDE	MLE Ave Dist	MLE StDev Dist	MDE Ave Dist	MDE StDev Dist	Benton-Santo Moment Ave Dist	Benton-Santo MDE Ave Dist
GGD9(0,0.5,1,1, 0,1,1,1, 0.25) Equivalent to 0.25 Expo(0.5) + 0.75 Expo(1)	750	50.4%	0.00091	0.00112	0.00090	0.00110	0.101567	0.105535
GGD9(0,0.5,1,1;0,1,1,1;0.5) Equivalent to 0.50 Expo(0.5) + 0.50 Expo(1)	750	52.3%	0.00038	0.00042	0.00037	0.00041	0.0862087	0.0910827
GGD9(0,0.5,1,1; 0,1,1,1; 0.75) Equivalent to 0.75 Expo(0.5) + 0.25 Expo(1)	750	51.4%	0.00048	0.00083	0.00048	0.00081	-	-
GGD9(0,2,1,1, 0,0.5,1,1, 0.25) Equivalent to 0.25 Expo(2) + 0.75 Expo(0.5)	750	54.6%	0.00138	0.00193	0.00128	0.00164	0.0134047	0.0159149
GGD9(0,2,1,1, 0,0.5,1,1, 0.5) Equivalent to 0.50 Expo(2) + 0.50 Expo(0.5)	750	52.7%	0.00116	0.00148	0.00113	0.00143	0.0208788	0.0281973
GGD9(0,2,1,1, 0,0.5,1,1, 0.75) Equivalent to 0.75 Expo(2) + 0.25 Expo(0.5)	750	51.5%	0.00123	0.00143	0.00118	0.00136	0.0339208	0.0428782
GGD9(0,3,1,1, 0,0.5,1,1, 0.25) Equivalent to 0.25 Expo(3) + 0.75 Expo(0.5)	750	52.1%	0.00111	0.00158	0.00105	0.00132	0.0073279	0.011018
GGD9(0,3,1,1, 0,0.5,1,1, 0.5) Equivalent to 0.50 Expo(3) + 0.50 Expo(0.5)	750	52.2%	0.00129	0.00147	0.00127	0.00145	0.136226	0.0265675
GGD9(0,3,1,1, 0,0.5,1,1, 0.75) Equivalent to 0.25 Expo(3) + 0.75 Expo(0.5)	750	55.7%	0.00118	0.00129	0.00113	0.00128	0.0235877	0.0381142

Fixed Parameters:

Mixed Exponential; C1=0, B1=1, P1=1, C2=0, B2=1, P1=1

Parameters fixed by MDE:

Mixed Exponential; M

Benton-Santo used Method of Moments instead of MLE. She used 500 replications and only calculated the Mean Square Error for the mixing proportion estimates. Her distance estimate was defined as

$$MSE = \frac{1}{500}\sum_{i=1}^{500}(m_i - m)^2 \quad (1:29,33).$$

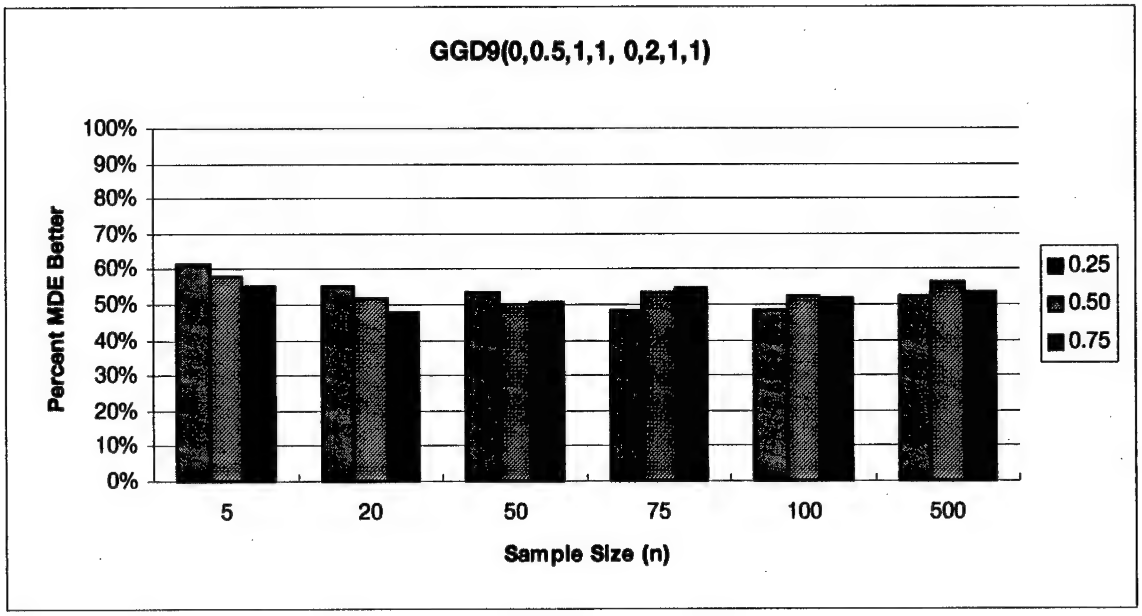

Figure 6 Percent MDE Better for Mixture of Exponentials with Known Locations

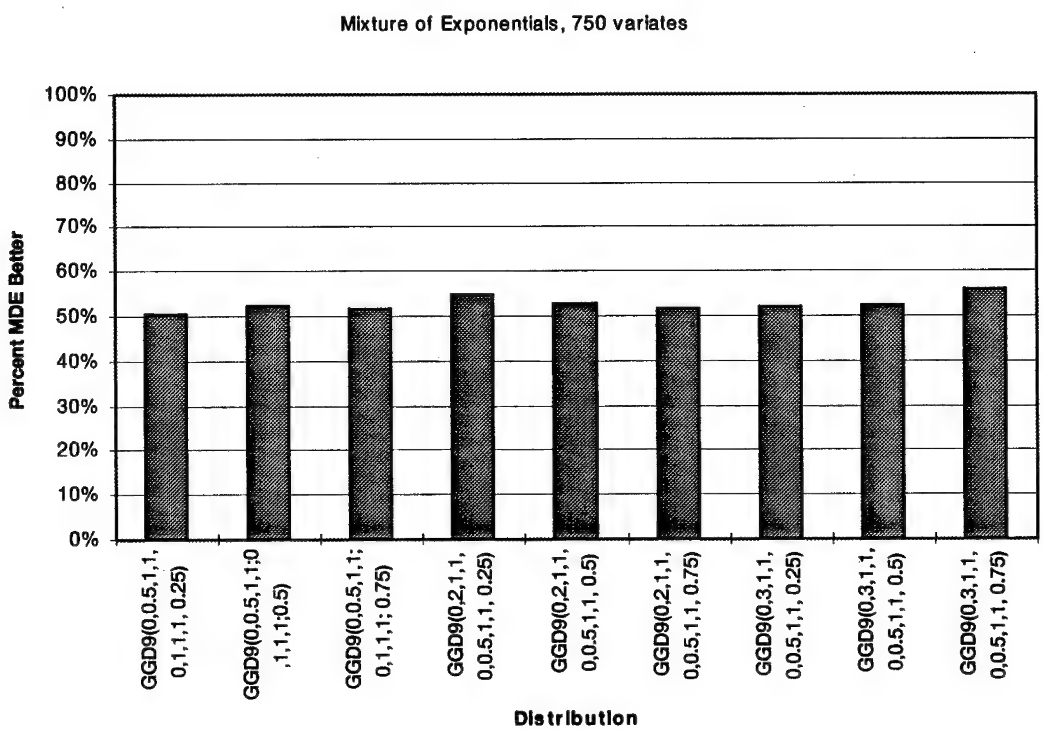

Figure 7 Percent MDE Better for Mix of Exponentials, 750 Variates

4.4 Mixture of Weibull Distributions with Known Locations Results

For the case of the mixture of Weibull distributions with known location parameters, it was assumed that the functional form of the distribution was known. Therefore, the shape/power parameters b1 and b2 were fixed at 1. A Maximum Likelihood Estimate was calculated. The mixture parameter was fixed using Minimum Distance and the remaining parameters re-estimated using Maximum Likelihood. Although a penalty function was tested to force the derivatives of the log likelihood function to zero, it did not appear to help with the estimates, thus the estimates were calculated without a penalty function.

Average distances and standard deviations of distance decreased for both Maximum Likelihood and for Minimum Distance as sample size increased. Minimum Distance showed that for this distribution it improved the estimates for all sample sizes for a large percentage of the replications run. The results are summarized in Table 14 and Figure 8.

Table 14 Mixture of Weibull Distributions, Known Locations, 1000 Replications

Distribution	*n*	*%MDE*	*MLE Ave Dist*	*MLE Stdev Dist*	*MDE Ave Dist*	*MDE Stdev Dist*
GGD9(0,4,1,0.5, 0,1,1,0.5, 0.25)	5	56.0%	0.8419	5.5058	0.5126	2.5931
Equivalent to	20	69.3%	0.0930	0.6888	0.0493	0.2710
0.25 Weibull(0.5,4) +	50	71.4%	0.0279	0.1099	0.0144	0.0560
0.75 Weibull(0.5,1)	75	67.6%	0.0239	0.2033	0.0124	0.1177
	100	69.1%	0.0188	0.1724	0.0109	0.1414
	500	63.7%	0.0021	0.0057	0.0014	0.0038
GGD9(0,4,1,0.5, 0,1,1,0.5, 0.5)	5	60.0%	0.7112	4.5078	0.4976	3.0340
Equivalent to	20	74.6%	0.1358	0.8827	0.0603	0.7216
0.50 Weibull(0.5,4) +	50	71.3%	0.0180	0.0533	0.0107	0.0278
0.50 Weibull(0.5,1)	75	68.7%	0.0160	0.0486	0.0081	0.0202
	100	68.9%	0.0114	0.0621	0.0070	0.0575
	500	64.5%	0.0019	0.0038	0.0014	0.0068
GGD9(0,4,1,0.5, 0,1,1,0.5, 0.75)	5	65.4%	1.1710	8.3871	0.9754	10.2107
Equivalent to	20	74.9%	0.0894	0.6536	0.0412	0.2613
0.25 Weibull(0.5,4) +	50	71.9%	0.0221	0.0804	0.0095	0.0176
0.75 Weibull(0.5,1)	75	70.6%	0.0197	0.1580	0.0081	0.0320
	100	70.6%	0.0128	0.0749	0.0059	0.0274
	500	62.5%	0.0015	0.0031	0.0009	0.0014

Fixed Parameters:

Mixed Weibull; C1=0, B1=1, C2=0, B2=1

Parameters fixed by MDE:

Mixed Weibull; M

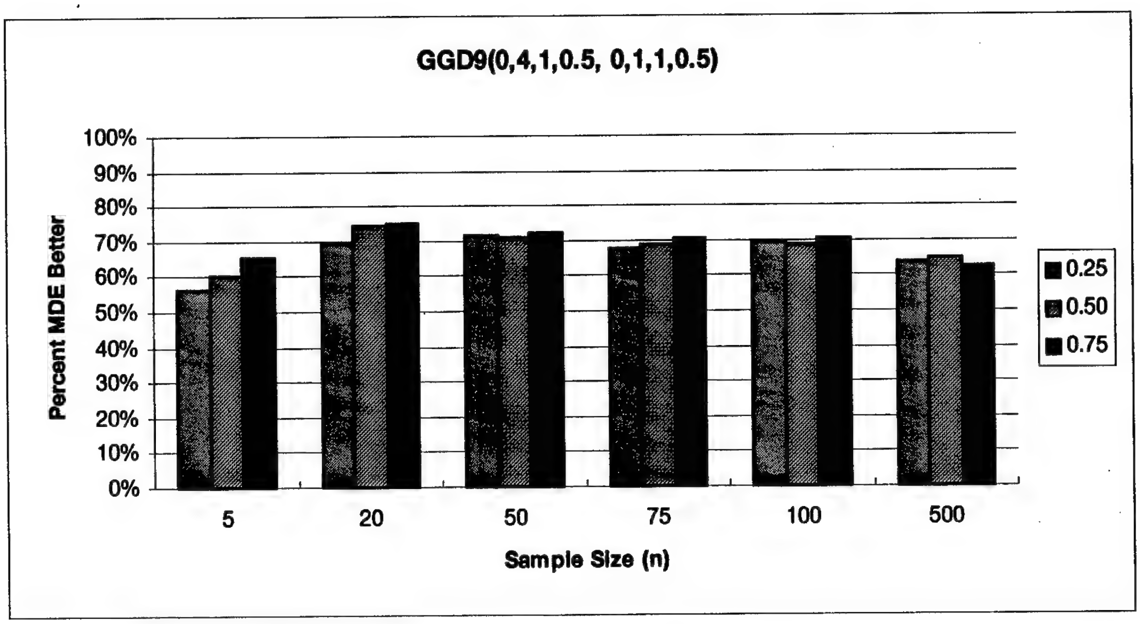

Figure 8 Percent MDE Better for Weibulls with Known Locations

4.5 Mixture of Exponentials with Unknown Locations Results

For the case of mixtures of Exponentials with unknown locations, it was assumed that the functional form of the distribution was known. Therefore, B1, P1, B2, P2 were all fixed at one. A Maximum Likelihood Estimate was found. The mixture parameter was fixed using Minimum Distance Estimation. Next, the smaller location parameter was fixed using Minimum Distance. The remaining parameters were then re-estimated.

Average Distance and standard deviation for both estimation techniques decreased with increasing sample size. Minimum Distance showed significant improvement in sample sizes of 5. It provided no significant improvements for sample sizes larger than that. The results are summarized in Table 15 and Figure 9.

Table 15 Mixture of Exponentials with Unknown Location Parameters, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD9(5,0.5,1,1, 10,0.5,1,1, 0.25)	5	64.0%	2.2724	7.6746	1.1175	7.5779
Equivalent to:	20	37.6%	0.1618	0.1531	0.2599	0.2006
0.25 Expo(0.5,5)	50	45.1%	0.0927	0.0709	0.1081	0.0982
+ 0.75 Expo(0.5,10)	75	45.9%	0.0811	0.0620	0.0867	0.0730
	100	45.5%	0.0797	0.0604	0.0797	0.0667
	500	50.2%	0.0642	0.0517	0.0529	0.0601
GGD9(5,0.5,1,1, 10,0.5,1,1, 0.5)	5	80.2%	1.4787	3.4304	0.4949	1.1731
Equivalent to:	20	30.5%	0.1354	0.1008	0.2278	0.1374
0.50 Expo(0.5,5)	50	33.8%	0.0494	0.0424	0.0899	0.0922
+ 0.50 Expo(0.5,10)	75	32.7%	0.0330	0.0383	0.0577	0.0632
	100	37.2%	0.0302	0.0353	0.0462	0.0503
	500	43.6%	0.0181	0.0296	0.0215	0.0361
GGD9(5,0.5,1,1, 10,0.5,1,1, 0.75)	5	82.8%	1.3638	2.2765	0.4414	0.4302
Equivalent to:	20	28.3%	0.1348	0.1169	0.2171	0.1114
0.75 Expo(0.5,5)	50	26.4%	0.0451	0.0406	0.1129	0.1011
+ 0.25 Expo(0.5,10)	75	23.5%	0.0260	0.0293	0.0767	0.0873
	100	28.7%	0.0188	0.0237	0.0487	0.0640
	500	37.1%	0.0157	0.0307	0.0191	0.0277

Fixed Parameters:

Mixed Exponential; B1=1, P1=1, B2=1, P1=1

Parameters fixed by MDE:

Mixed Exponential; M, C1

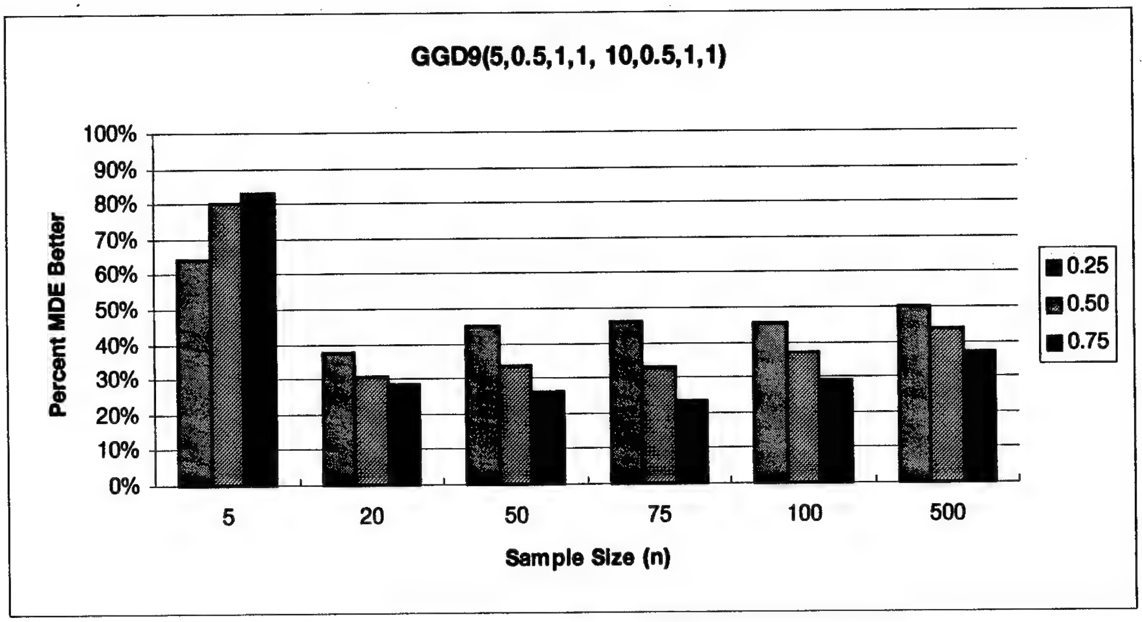

Figure 9 Percent MDE Better for Mixture of Exponentials, Unknown Locations

4.6 Mixture of Weibull Distributions with Unknown Locations Results

For the case of mixture of Weibull distributions with unknown location parameters, it was assumed that the functional form of the distribution was known. Therefore, B1 and B2 were fixed at one. The other parameters were estimated using MLE. The mixture parameter was fixed using Minimum Distance Estimation, then the smaller location parameter was fixed using MLE. The remaining parameters were re-estimated using MLE.

Three types of mixtures of Weibull distributions were estimated. The first was a widely separated mixture of two Weibull distributions with power parameter less than one. The second was a widely separated mixture of two Weibull distributions with power parameter greater than one. The third was a non-separated mixture of two Weibull distributions with different scale parameters. For the widely separated mixtures, MDE improved the parameter estimations over MLE. Surprisingly, for some small sample sizes for Weibull distributions with power less than one, MDE did not appear to improve the estimation, whereas it did for higher sample sizes. For the mixture of two non-separated Weibull distributions, MDE did not appear to help improve the estimates too much. The results are summarized in Table 16, 17, and 18 and in Figure 10, 11 and 12.

Table 16 Mixture of Non-Separated Weibull Distributions, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*	*Mumford %MDE Better*
GGD9(5,4,1,0.5, 5,1,1,0.5, 0.1)	5	49.4%	13.4865	189.1606	9.5115	114.2607	-
Equivalent to	10	53.4%	8.7929	129.3095	4.6741	46.6752	53.5%
0.1 Weibull (0.5,4,5) +	20	60.8%	5.0829	57.0811	2.5644	32.4602	43.2%
0.9 Weibull (0.5,1,5)	50	58.5%	1.4275	9.2331	0.9035	6.3840	-
	75	50.9%	1.9690	26.1039	0.7328	7.3919	-
	100	49.2%	1.8295	19.5849	0.4764	2.3463	42.8%
GGD9(5,4,1,0.5, 5,1,1,0.5, 0.3)	5	46.2%	18.5703	502.2032	3.9883	16.4149	-
Equivalent to	10	52.1%	3.6368	48.8797	2.3246	6.8973	55.2%
0.3 Weibull (0.5,4,5) +	20	59.1%	1.7274	10.1159	1.5340	6.8951	55.6%
0.7 Weibull (0.5,1,5)	50	52.1%	1.2914	9.7886	0.4992	1.5360	-
	75	45.2%	17.4282	516.9004	0.9029	17.6258	-
	100	40.8%	0.7800	8.9131	0.4304	3.1905	56.3%
GGD9(5,4,1,0.5, 5,1,1,0.5, 0.5)	5	47.9%	16.8482	304.9816	4.0251	24.0798	-
Equivalent to	10	52.7%	1.2135	3.9290	2.3620	9.9836	93.2%
0.5 Weibull (0.5,4,5) +	20	57.0%	1.5634	14.2040	1.2442	5.6163	98.1%
0.5 Weibull (0.5,1,5)	50	45.7%	0.5765	4.0628	0.4739	1.8064	-
	75	41.7%	1.8968	23.1605	0.3687	1.1389	-
	100	41.3%	0.2024	1.0878	0.2801	0.5711	99.0%

Fixed Parameters:

Mixed Weibull; B1=1, B2=1

Parameters fixed by MDE:

Mixed Weibull; M, C1

Mumford used a variety of Minimum Distance settings (37:42-44,50).

Table 17 Mixture of Widely Separated Weibull Distributions with Power <1, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*	*Mumford %MDE Better*
GGD9(5,0.5,1,0.9, 10,0.5,1,0.9, 0.1)	5	59.5%	16.1714	108.6143	4.4572	13.9817	-
Equivalent to	10	84.5%	7.5789	40.5318	1.6038	9.9351	86.3%
0.1 Weibull (0.9,0.5,5) +	20	94.4%	8.2328	76.6810	0.6658	2.4105	82.0%
0.9 Weibull (0.9,0.5,10)	50	98.5%	7.9524	105.5130	0.5657	3.6631	-
	75	98.0%	17.3983	314.9763	0.5450	2.2690	-
	100	98.4%	8.1671	89.3714	0.4679	1.1362	84.6%
GGD9(5,0.5,1,0.9, 10,0.5,1,0.9, 0.3)	5	49.5%	28.7628	179.1810	5.8101	23.8569	-
Equivalent to	10	65.8%	7.9255	40.4792	2.0909	7.6177	41.3%
0.3 Weibull (0.9,0.5,5) +	20	76.6%	34.5906	808.7572	1.1381	4.5695	45.9%
0.7 Weibull (0.9,0.5,10)	50	89.5%	9.1949	101.6483	0.6468	2.8552	-
	75	92.0%	4.6411	28.8257	0.6453	3.4988	-
	100	93.4%	4.6851	27.5269	0.5303	3.5554	51.9%
GGD9(5,0.5,1,0.9, 10,0.5,1,0.9, 0.5)	5	44.6%	18.1195	136.0094	6.2287	19.1815	-
Equivalent to	10	49.0%	8.4700	43.1619	3.9416	22.3347	86.3%
0.5 Weibull (0.9,0.5,5) +	20	57.3%	7.1467	56.5441	1.5195	3.2892	82.0%
0.5 Weibull (0.9,0.5,10)	50	65.9%	10.2861	139.5968	2.3714	30.1318	-
	75	67.0%	16.0948	232.5186	0.7965	1.8866	-
	100	67.6%	9.2925	78.4555	0.9382	6.0059	84.6%

Fixed Parameters:

Mixed Weibull; B1=1, B2=1

Parameters fixed by MDE:

Mixed Weibull; M, C1

Mumford used a variety of Minimum Distance settings (37:42-44,52).

Table 18 Mixture of Widely Separated Weibull Distributions with Power >1, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*	*Mumford %MDE Better*
GGD9(5,0.5,1,3, 10,0.5,1,3, 0.1)	5	80.0%	10.5399	134.3711	2.3503	13.4631	-
Equivalent to	10	74.6%	5.8005	64.5266	0.9938	3.3484	60.9%
0.1 Weibull (3,0.5,5) +	20	78.3%	4.3341	48.4001	0.7052	0.8290	44.2%
0.9 Weibull (3,0.5,10)	50	82.4%	3.6058	37.0374	0.6286	0.7245	-
	75	84.3%	10.7920	164.8084	0.5922	0.4017	-
	100	85.4%	15.8059	274.6573	0.5867	0.7261	28.4%
GGD9(5,0.5,1,3, 10,0.5,1,3, 0.3)	5	68.4%	40.1347	662.1318	50.5883	1423.6050	-
Equivalent to	10	74.9%	5.0686	72.9993	1.8382	9.2338	41.6%
0.3 Weibull (3,0.5,5) +	20	76.0%	3.0514	29.3728	1.1222	3.7143	29.0%
0.7 Weibull (3,0.5,10)	50	85.7%	22.0531	602.9458	0.6748	0.9315	-
	75	79.4%	8.9066	58.4360	4.0360	26.7977	-
	100	87.0%	15.1667	196.6024	0.6447	0.7103	39.6%
GGD9(5,0.5,1,3, 10,0.5,1,3, 0.5)	5	68.1%	18.7441	286.7962	2.9921	11.6497	-
Equivalent to	10	77.7%	4.3577	47.6659	1.9486	7.5849	82.8%
0.5 Weibull (3,0.5,5) +	20	83.9%	4.3289	57.9696	0.9862	2.3134	86.9%
0.5 Weibull (3,0.5,10)	50	88.1%	2.3438	21.3958	0.7094	2.3961	-
	75	89.5%	4.8142	64.7030	28.0315	866.4996	-
	100	90.8%	14.5109	265.0473	0.5774	0.6980	94.7%

Fixed Parameters:

Mixed Weibull; B1=1, B2=1

Parameters fixed by MDE:

Mixed Weibull; M, C1

Mumford used a variety of Minimum Distance settings (37:42-44,51).

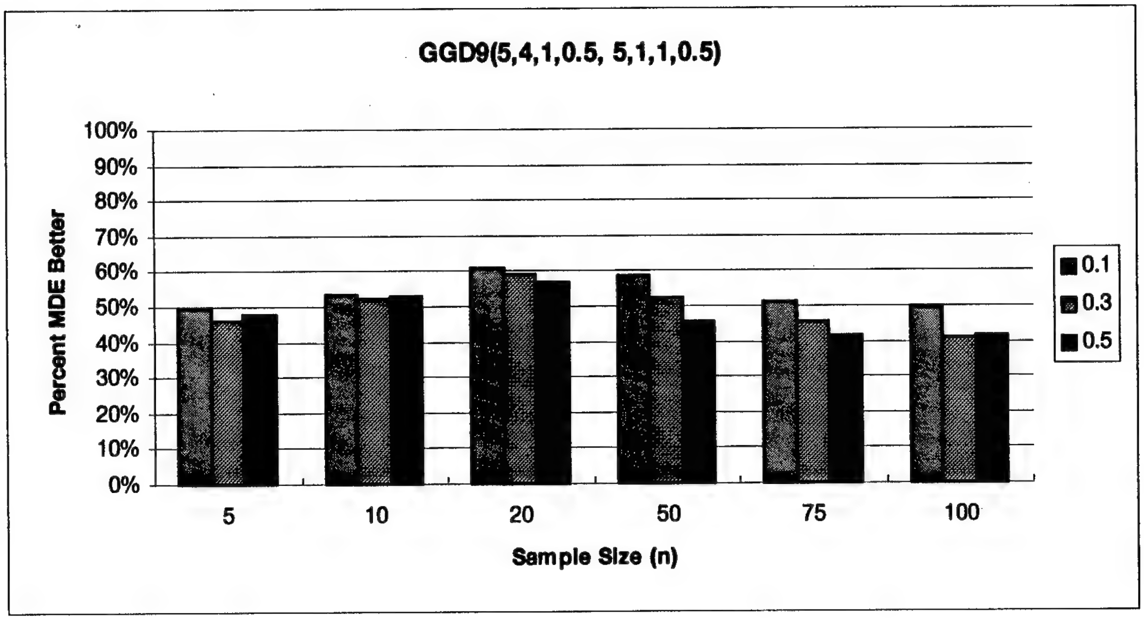

Figure 10 Percent MDE Better for Mixture of Non-Separated Weibulls, Unknown Locations

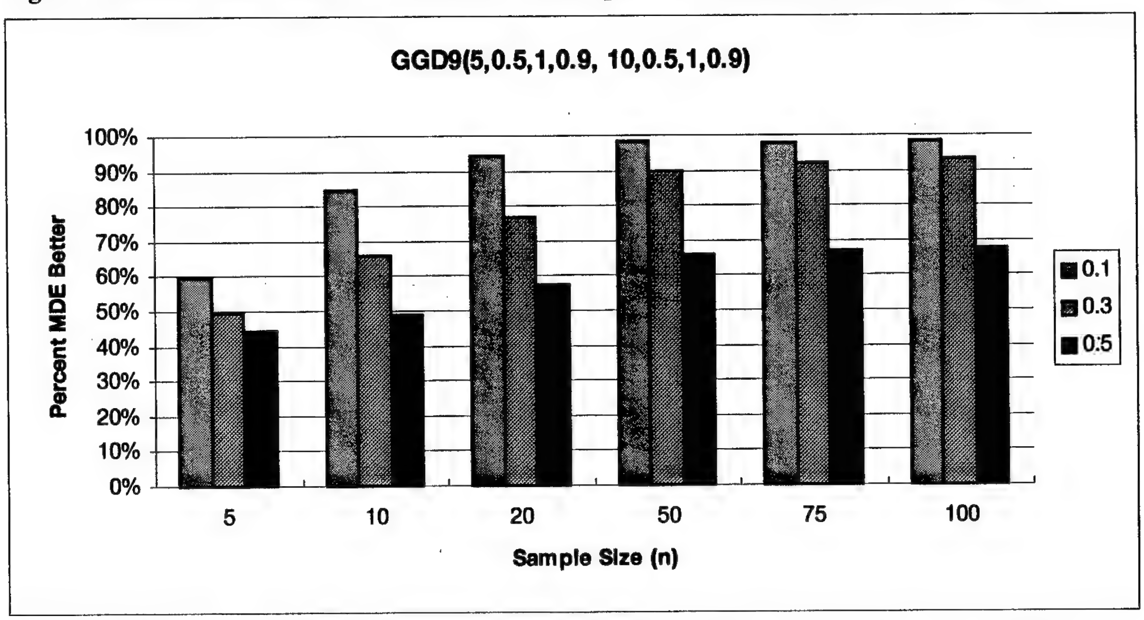

Figure 11 Percent MDE Better for Widely Separated Weibulls, Power < 1, Unknown Locations

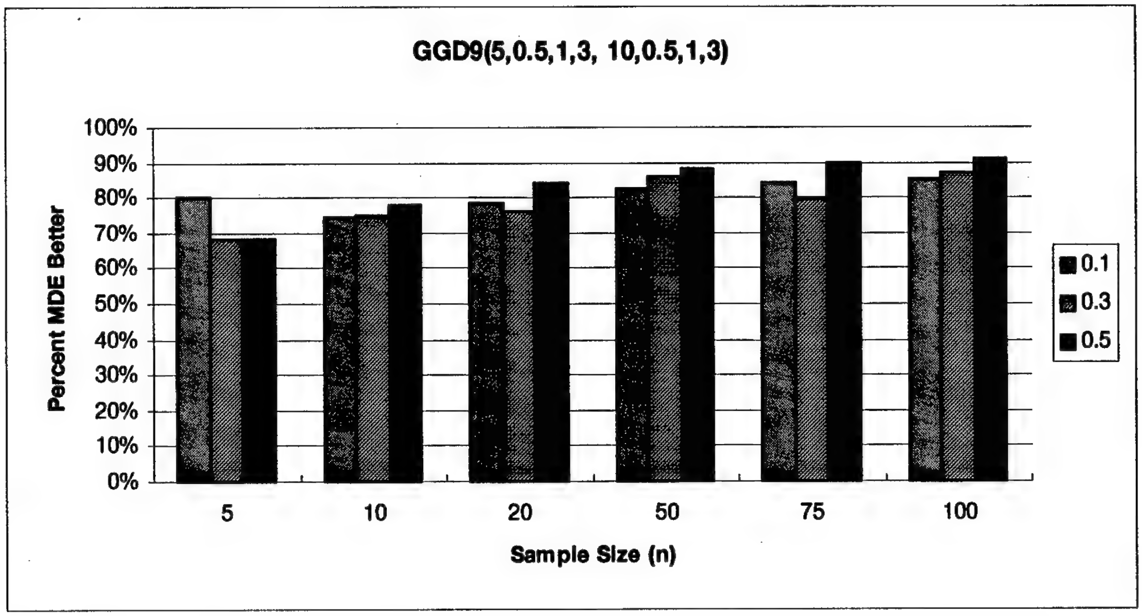

Figure 12 Percent MDE Better for Widely Separated Weibulls Power > 1, Unknown Locations

4.7 Mixture of Generalized Gamma Distribution Results

For the case of the mixture of Generalized Gamma Distributions. It was assumed that the functional form of the distribution was known. No parameters were fixed prior to estimation All the parameters were estimated using MLE. The mixture parameter was fixed using Minimum Distance, and then the smaller location parameter was fixed using Minimum Distance. The remaining parameters were re-estimated using MLE.

Three types of mixtures of the Generalized Gamma Distributions were tested. The first was a mixture of two half-normal distributions. The second was a mixture of gamma distributions. The third was mixture of Weibull distributions. Average distances and standard deviations for each of these cases tended to decrease with sample size, but this was not always the case, because there were some extreme outliers. For the most part, Minimum Distance did not improve parameter estimates. This is caused by the fact that the MLE estimates were much further from the true populations than in previous cases. The results are summarized in Table 19, 20 and 21, as well as in Figure 13, 14 and 15.

An attempt was made to improve the Maximum Likelihood estimate by restarting the Genetic Algorithm 15 times for the estimation of the GGD9(5, 0.5,1,2, 10,0.5,1,2 0.5) which is equivalent to 0.5 Weibull (2,0.5,5) + 0.5 Weibull(2,0.5,10) for sample sizes equal to ten. This was attempted so that the Maximum Likelihood estimate would not be affected by the initial population draws of the Genetic Algorithm. It did not improve the Maximum Likelihood Estimate.

Table 19 Mixture of GGD9, Mix of Half-Normal Distributions Results, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD9(5,0.5,0.5,2, 10,0.5,0.5,2, 0.1)	5	36.5%	17.2074	235.0678	14.7188	173.3774
Equivalent to	10	42.7%	160.2340	4065.4007	214.2402	6353.3455
0.1 Half-Normal(0.5,5) +	20	41.4%	24.1075	317.4191	5.4549	48.8895
0.9 Half-Normal(0.5,10)	50	35.4%	9.2536	55.6258	14.6415	342.8181
	75	30.1%	35.3408	800.1484	4.1229	19.4784
	100	28.8%	12.0450	103.1655	16.2656	412.0095
GGD9(5,0.5,0.5,2, 10,0.5,0.5,2, 0.3)	5	35.9%	172.1579	4902.1669	25.8920	699.5237
Equivalent to	10	32.5%	57.4000	1101.7336	29.9420	764.9027
0.3 Half-Normal(0.5,5) +	20	31.7%	33.1320	655.6209	8.2653	63.0886
0.7 Half-Normal(0.5,10)	50	30.1%	233.4613	4247.8926	8.9785	107.4095
	75	30.1%	34.7074	744.7061	7.3780	68.8707
	100	31.5%	97.7347	2680.0596	12.8910	132.3850
GGD9(5,0.5,0.5,2, 10,0.5,0.5,2, 0.5)	5	22.8%	16.5445	300.2714	2.2366	10.5080
Equivalent to	10	25.0%	13.8057	122.0543	22.0262	438.7600
0.5 Half-Normal(0.5,5) +	20	29.0%	195.1772	5736.0197	8.9189	77.0940
0.5 Half-Normal(0.5,10)	50	31.0%	11.1105	60.9907	24.3062	353.2711
	75	31.6%	21.0423	310.3885	44.9672	816.7931
	100	32.8%	9.5300	90.2882	15.1278	157.5469

Fixed Parameters:

Mixed Generalized Gamma; none

Parameters fixed by MDE:

Mixed Weibull; M, C1

Table 20 Mixture of GGD9, Mix of Weibull Distributions Results, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD9(5,0.5,1,2, 10,0.5,1,2, 0.1)	5	39.2%	325.4816	3041.3151	417.9064	7330.9237
Equivalent to	10	36.1%	83.6890	709.4128	102.1803	2929.3164
0.1 Weibull(2,0.5,5) +	20	40.1%	2049.5534	55100.3625	10.8364	132.0712
0.9 Weibull(2,0.5,10)	50	46.5%	141.1049	2016.3228	5.7318	45.5691
	75	51.6%	276.7355	7004.7223	9.3178	114.8154
	100	56.2%	43.2844	452.0912	20.0893	371.8290
GGD9(5,0.5,1,2, 10,0.5,1,2, 0.3)	5	23.1%	48.1144	550.5923	14.5816	161.7734
Equivalent to	10	26.9%	46.4158	693.9520	19.1025	330.2501
0.3 Weibull(2,0.5,5) +	20	35.3%	117.4913	2331.9271	15.9278	179.7710
0.7 Weibull(2,0.5,10)	50	43.4%	64.7897	1028.8367	53.1290	1188.0150
	75	48.6%	379.4672	8239.3617	15.6233	114.6346
	100	48.8%	90.4284	1598.1520	11.3831	65.2074
GGD9(5,0.5,1,2, 10,0.5,1,2, 0.5)	5	20.5%	76.9953	1566.1730	16.7244	134.9463
Equivalent to	10	23.6%	54.1197	687.8037	124.5002	3058.3391
0.5 Weibull(2,0.5,5) +	20	32.3%	99.8584	2306.5141	16.9092	188.1700
0.5 Weibull(2,0.5,10)	50	38.8%	206.1879	5423.9182	318.4078	9035.1012
	75	26.0%	76.6731	810.6282	56.9058	832.2582
	100	42.7%	105.9541	1635.9082	28.3106	228.3784

Fixed Parameters:

Mixed Generalized Gamma; none

Parameters fixed by MDE:

Mixed Weibull; M, C1

Table 21 Mixture of GGD9, Mix of Gamma Distributions, Results, 1000 Replications

Distribution	*n*	*%MDE Better*	*MLE Ave Dist*	*MLE StDev Dist*	*MDE Ave Dist*	*MDE StDev Dist*
GGD9(5,0.5,2,1, 10,0.5,2,1, 0.1)	5	23.0%	3839.5630	14198.2036	30965.6477	107927.1025
Equivalent to	10	29.8%	818.9738	3836.4657	22770.1562	97998.8013
0.1 Gamma(0.5,2,5) +	20	37.0%	253.6583	2941.2262	8244.0625	56846.6629
0.9 Gamma(0.5,2,10)	50	44.3%	1286.4060	38047.1286	105.9396	3042.7405
	75	47.2%	90.1245	867.8399	17.8362	214.2934
	100	47.1%	31.8820	312.7171	5.6627	48.8987
GGD9(5,0.5,2,1, 10,0.5,2,1, 0.3)	5	20.3%	426.5057	4244.6433	2943.3344	20946.5659
Equivalent to	10	28.0%	115.0250	1014.4080	470.7710	6819.6270
0.3 Gamma(0.5,2,5) +	20	34.8%	169.4732	2068.7955	26.8496	277.0682
0.7 Gamma(0.5,2,10)	50	43.4%	134.8274	2012.9430	43.0410	532.2295
	75	44.7%	143.5685	2194.8084	74.5137	1723.2606
	100	46.7%	191.0620	3022.1309	12.5361	148.4399
GGD9(5,0.5,2,1, 10,0.5,2,1, 0.5)	5	16.7%	212.9728	4808.2843	868.0391	24436.3313
Equivalent to	10	26.3%	88.9492	1134.4886	63.3647	781.1593
0.5 Gamma(0.5,2,5) +	20	32.8%	29.7252	401.6574	13.9387	154.2059
0.5 Gamma(0.5,2,10)	50	35.5%	67.2405	647.6326	95.4062	1940.4139
	75	41.1%	347.8163	5096.1299	75.5559	1043.2663
	100	42.1%	166.4310	2298.5811	152.5097	3295.3148

Fixed Parameters:

Mixed Generalized Gamma; none

Parameters fixed by MDE:

Mixed Weibull; M, C1

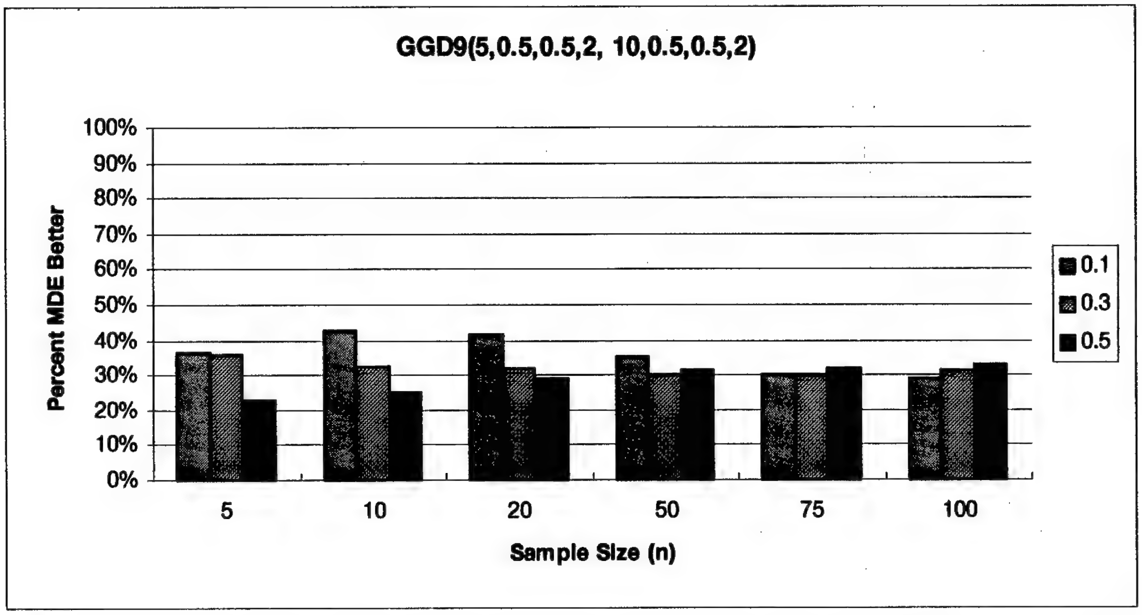

Figure 13 Percent MDE Better, GGD9, Mixture of Half-Normal Distributions

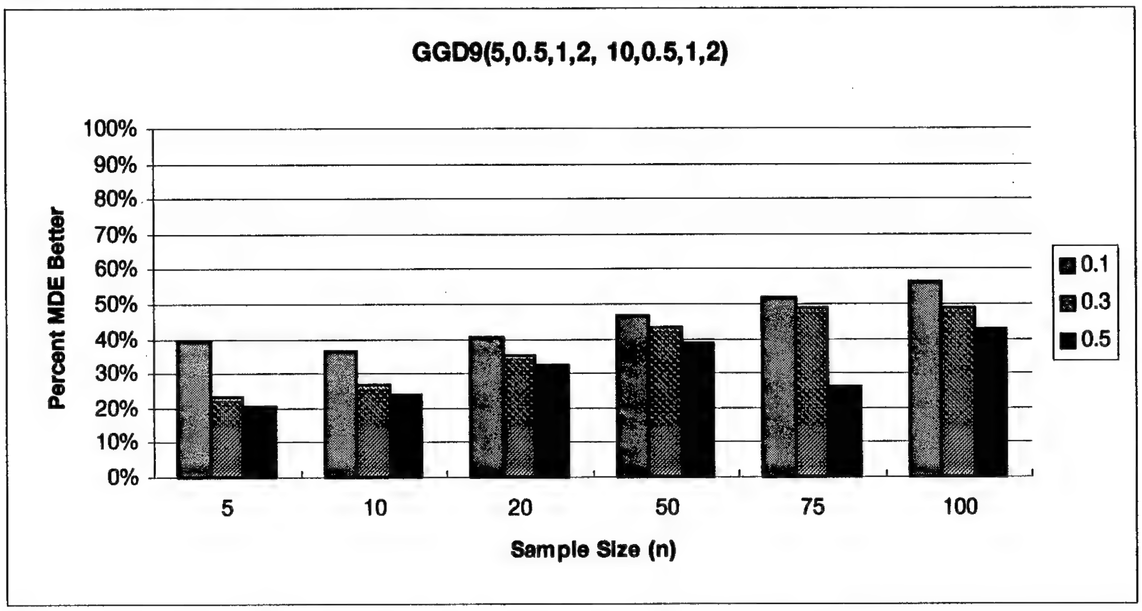

Figure 14 Percent MDE Better, GGD9, Mixture of Weibull Distributions

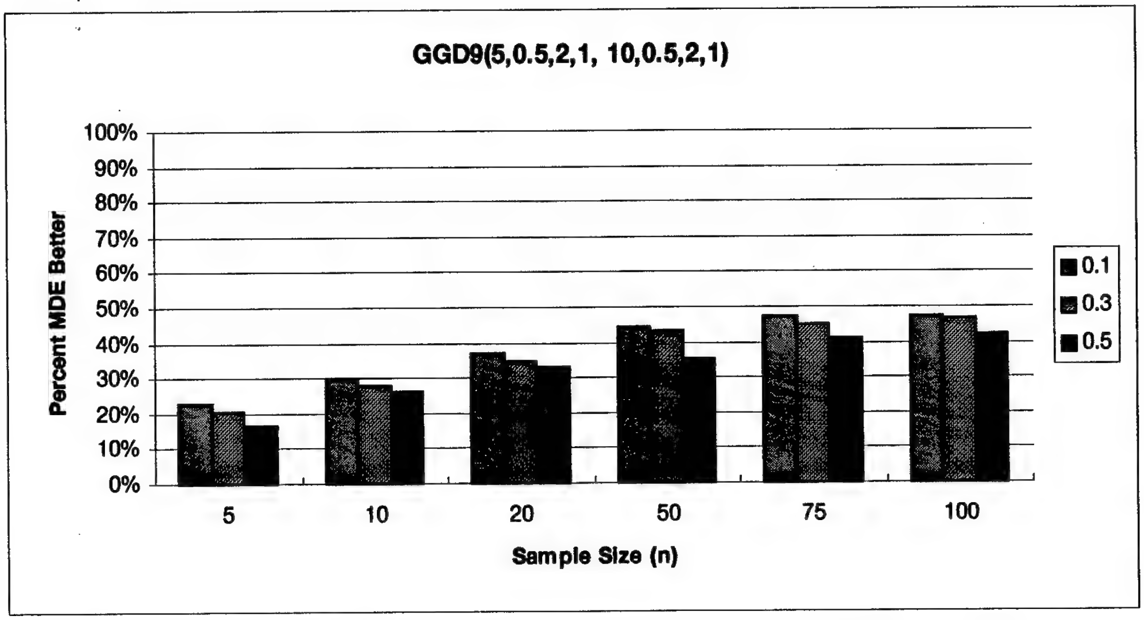

Figure 15 Percent MDE Better, GGD9, Mixture of Gamma Distributions

V. Conclusions

5.1 MLE Conclusions

Maximizing the Maximum Likelihood Equation using a Genetic Algorithm worked well but gave poorer estimates as the number of parameters being estimated got larger, particularly when both location and mixture parameters were being estimated. For the most part, it behaved exactly as expected; as sample size grew, the average distance and standard deviation of the distance decreased. It tended to improve relative to Minimum Distance as sample size grew larger. It provided excellent estimators for the single component parameters. The use of the first derivative penalty function worked very well for the single component distributions. It, also, provided good estimators for the mixture of Exponential distributions with both known and unknown location parameters, and for the mixture of Non-separated Weibull distributions with known location parameters without the use of any first derivative penalty functions. MLE did not work as well for the mixture of Weibull distributions with unknown location parameters and did particularly poorly for the mixture of Generalized Gamma Distributions. The use of first derivative penalty functions for the mixture distributions did not help in this research, so it was abandoned, but with the proper scaling it might work. Mumford's technique obtained lower MSE than this research. His use of the first derivative information for the Mixture of Weibull distributions with unknown location parameter gives evidence that it works. There must be a way to use that information as well for a Genetic Algorithm.

5.2 MDE Conclusions

Minimum Distance provided an improvement in many of the cases tested, particularly for small sample sizes. It improved the estimates of the Maximum Likelihood Estimator for the single component distributions, for the mixture of Exponential distributions with known parameters and for the mixture of non-separated Weibull distributions with known location parameters. It did not provide better estimates for the Mixture of Weibull distributions with unknown location parameters or for the Mixture of Generalized Gamma Distributions. These are the same distributions that the Maximum Likelihood Estimators did not

provide as good as estimates for. It is known that Minimum Distance is sensitive to the initial estimate that it is provided. For the cases where it did not receive a good initial estimate, it performed poorly. It was hoped to show that Minimum Distance would improve parameter estimation for the Mixed Generalized Gamma Distribution. This was not shown because the Maximum Likelihood Estimates were so poor.

5.3 Recommendations

There are several avenues for continued research that this work has shown. First of all, the possibility of using first derivative information for the parameter estimation of the Mixed Generalized Gamma Distribution still exists. Given that the proper scaling can be discovered, much better MLE parameters could be calculated and thereby also improve the initial estimate for Minimum Distance, which would then improve its estimation capability, possibly to the point that it improves relative to Maximum Likelihood Estimation.

Minimum Distance could be applied to other parameters not covered in this research. The large distances seen when estimating the parameters for mixture of Generalized Gamma Distributions might be reduced by using Minimum Distance on the power/shape parameters b1 and b2. These estimated power/shape parameters were far from their true parameters, and were forcing the other parameters to compensate for them. This had the unfortunate effect of driving the other estimated parameters far from their true parameters.

Another problem that was encountered was that the second location parameter, c_2, was being pushed above the highest variates in the sample and the mixture parameter was nearing one. This means that the estimation techniques were trying to fit the mixture with a single distribution. This research penalized the objective if c_2 was greater than the largest variate in the sample. Further research could determine if a different upper limit for it would improve its distance from true such as the k^{th} order statistic vs. the n^{th} order statistic where $n > k > 1$.

Another possibility for improving the efficiency of this Genetic Algorithm used is to improve the stopping criteria. This research used a stopping criteria that perhaps was too simple. It simply checked every 200^{th} generation for MLE and every 150^{th} for MDE and then stopped if the best individual did not

change in that amount of time. The number of generations run could possibly be reduced by a stopping criteria based on percentage improvements in the best individual over successive generations. It could also be reduced by storing the values of the 200 previous generations and checking the stopping criteria every generation. This could greatly reduce the number of function evaluations that are required, which would give significant reduction in simulation run times.

The results tables showed average distance and average standard deviation of distance. These measures can be greatly affected by outliers. Other measures that are less sensitive to outliers, such as the median of the distance should be considered. Study of outliers should be considered in any future research that will use the distance as a measure of how much Minimum Distance improves the parameter estimates of Maximum Likelihood.

Appendix A PDFs for Special Case Distributions

Key for Special Case Distributions taken from Law & Kelton, (location parameter added) (31:331-335) :

Exponential: Expo(b,c)

$$f(x;b,c) = \frac{1}{b} \cdot e^{-(x-c)/b} \qquad x > c > 0$$

Expo(b) = Expo(b,0) = GGD4(0,b,1,1)

Gamma: Gamma(a,b,c)

$$f(x) = \frac{(x-c)^{a-1} \cdot e^{-(x-c)/b}}{\Gamma(a)} \qquad x > c > 0$$

Gamma(a,b) = Gamma(a,b,0) = GGD4(0,a,b,1)

Weibull: Weibull(a,b,c)

$$f(x) = \frac{a \cdot (x-c)^{a-1} \cdot e^{-\{(x-c)/b\}^a}}{b^a} \qquad x > c > 0$$

Weibull(a,b) = Weibull(a,b,0) = GGD4 (0,b,1,a)

Generalized Gamma: GGD4(c,a,b,p)

$$f(x;c,a,b,p) = \frac{p \cdot (x-c)^{b \cdot p-1} \cdot e^{-[(x-c)/a]^p}}{a^{b \cdot p} \cdot \Gamma(b)}$$

where a, b, p ≥ 0 and $x > c \geq 0$ (11:2).

GGD3(a,b,p) = GGD4(0,a,b,p)

Appendix B Source Code

Notes to the Code

1. Single quotes (') denote comments which are not executed.

2. All public, i.e. global or common, variables are designated with a "pv" .

3. Subroutine calls have been documented as best possible to give the module that the called routine can be found in. If no module reference is given, the called subroutine will be in the same module as the calling subroutine.

4. The line continuation character "_" when used is found one space after the last text on a line. It similar to the "&" in card column 6 of FORTRAN. It continues the next line as part of the previous line.

Public & Settings Module

```
Option Explicit
'Contains variables and settings used throughout the program.


Public Const pvPopSize = 5         'Max number of individual in a generation.
Public Const pvSmallestParam = 0.0078125   'Smallest value a parameter can be
Public Const pvBiggestParam = 15.9921875   'Biggest value a parameter can be

Public pvNumMutation    As Long    'Number of mutations that occur
Public pvNCross         As Long    'Number of crossovers that occur

Public pvMDE            As Boolean         'True if MDE, false if on mle

Public pvMaxVariates    As Integer
Public pvCheck          As Integer         'check to see Every pvCheck for convergence.
Public pvLChrom         As Integer         'Length of a Chromosome

Public pvMaxLocParam    As Single          'This is the biggest the LocParam can ever be.
Public pvC1             As Single          'This is the smaller location parameter
Public pvC2             As Single          'This is the bigger location parameter
Public pvM              As Single          'This is the mixture parameter
Public pvBiggestVariate As Single          'This is the nth order statistic.

Type IndividualRecord
   Chrom(76)            As Boolean         ' Each position is an Allele: True=1,False=0
   C1                   As Single
   A1                   As Single
   B1                   As Single
   P1                   As Single
   C2                   As Single
   A2                   As Single
   B2                   As Single
   P2                   As Single
   M                    As Single
   Fitness              As Double         ' Objective function value
   Parent1              As Integer       ' parents
   Parent2              As Integer
   Xsite                As Integer       'cross point
End Type

Type Population
    Individual(pvPopSize) As IndividualRecord
End Type
```

```
'
' This initializes the settings for Maximum Likelihood Estimation
'
Sub SetupForMLE()
   pvMDE = False
   pvLChrom = 76
   pvCheck = 200
End Sub

'
' This initializes the settings for Minimum Distance Estimation
'
Sub SetupForMDE()
   pvMDE = True
   pvLChrom = 60
   pvCheck = 150
End Sub
```

Driver Module

```
Option Explicit
' Contains the Driver Routine.


'
' Runs the GA for different sample sizes and writes them
' to separate worksheets.
Sub Driver()

  pvMaxVariates = 50
  Call RunGA

  Sheets("Output").Copy Before:=Sheets(1)
  Sheets("Output").Range("a:iv").ClearContents
  pvMaxVariates = 5
  Call RunGA

  Sheets("Output").Copy Before:=Sheets(1)
  Sheets("Output").Range("a:iv").ClearContents
  pvMaxVariates = 20
  Call RunGA

  Sheets("Output").Copy Before:=Sheets(1)
  Sheets("Output").Range("a:iv").ClearContents
  pvMaxVariates = 50
  Call RunGA

  Sheets("Output").Copy Before:=Sheets(1)
  Sheets("Output").Range("a:iv").ClearContents
  pvMaxVariates = 75
  Call RunGA

  Sheets("Output").Copy Before:=Sheets(1)
  Sheets("Output").Range("a:iv").ClearContents
  pvMaxVariates = 100
  Call RunGA

End Sub
```

```
'
' This runs the MicroGA 1000 times and saves it periodically.
'
Sub RunGA()
   Dim Distrib    As String
   Dim i          As Integer
   Dim UppLim     As Single

On Error GoTo handleCancel
   Application.EnableCancelKey = xlErrorHandler

   Application.DisplayAlerts = False

   Sheets("Output").Select

   Const TC1 = 5        'True Parameters
   Const TA1 = 2
   Const TB1 = 2
   Const TP1 = 1

   Const TC2 = 10
   Const TA2 = 2
   Const TB2 = 2
   Const TP2 = 1

   Const TM = 0.5


   Distrib = "GGD9(" & TC1 & "," & TA1 & "," & TB1 & "," & TP1 & ", " & _
        TC2 & "," & TA2 & "," & TB2 & "," & TP2 & ", " & TM & ") n=" & pvMaxVariates

   Range("A1").NoteText Text:=Distrib, Start:=1
   Range("A1").Value = Distrib
   Range("a1").Select

  UppLim = FindUppLim(TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM, TC2) 'in Integrated Dist Mod

   For i = 1 To 1000
      Application.StatusBar = Range("A1").CurrentRegion.Rows.Count - 1
      Call MicroGA(TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM, UppLim)
      If (i Mod 48) = 0 Then
         ActiveWorkbook.Save
      End If
   Next i

 handleCancel:

 Sheets("Driver Mod").Select
 ActiveWorkbook.Save

 MsgBox "Data Saved. Select 'Driver' macro to Continue.", vbExclamation, "Dean's Thesis"
 End Sub
```

```
'
' ClearOutputSheet Macro
' Macro recorded 2/14/98 by Dean Boerrigter
' Removes all data from output sheet and formats it.
'
Sub ClearOutputSheet()

    Sheets("Output").Select
    Sheets("Output").Range("a:iv").ClearContents

    Columns("A:A").ColumnWidth = 3.57
    Columns("A:A").ColumnWidth = 4
    Columns("B:B").ColumnWidth = 5.86
    Columns("C:C").ColumnWidth = 35.57
    Columns("D:D").ColumnWidth = 38.43
    Columns("E:E").ColumnWidth = 11.29
    Columns("F:F").ColumnWidth = 11.29
    ActiveWindow.Zoom = 75

    Range("B2").Select
    ActiveWindow.FreezePanes = True
End Sub
```

Gen Gamma Module

```
Option Explicit
'Contains the Mixed Generalized Gamma Functions.


'
' Cumulative density function for the 9-parameter Generalized Gamma
' Distribution
'
Function GGD9cdf(X As Single, C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single,
A2 As Single, B2 As Single, P2 As Single, M As Single) As Double

  Dim cdf1 As Single
  Dim cdf2 As Single

  cdf1 = M * GGD4cdf(X, C1, A1, B1, P1)

  cdf2 = 0
  If X > C2 Then
     cdf2 = (1 - M) * GGD4cdf(X, C2, A2, B2, P2)
  End If

  GGD9cdf = cdf1 + cdf2
End Function


'
' Probability density function for the 9-parameter Generalized Gamma
' Distribution
'
Function GGD9pdf(X As Single, C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single,
A2 As Single, B2 As Single, P2 As Single, M As Single) As Single

  Dim pdf1 As Single
  Dim pdf2 As Single

  pdf1 = M * Exp(lnGGD4pdf(X, C1, A1, B1, P1))


  If X > C2 Then
     pdf2 = (1 - M) * Exp(lnGGD4pdf(X, C2, A2, B2, P2))
  Else
     pdf2 = 0
  End If

  GGD9pdf = pdf1 + pdf2
End Function
```

```
' This function returns the value of the
'  4-parameter Generalized Gamma function Cumulative Distribution Function
Function GGD4cdf(X As Single, c As Single, a As Single, b As Single, p As Single) As Single

   Dim IncompleteGamma          As Single
   Dim UpperLimit               As Double


   UpperLimit = ((X - c) / a) ^ p
   If UpperLimit > 1000000000000# Then
      GGD4cdf = 0.999999
      Exit Function
   End If


   IncompleteGamma = GaussLegendreQuadrature(0, UpperLimit, 10, b)
   GGD4cdf = IncompleteGamma / Exp(lnGamma(b))
End Function


' The natural log of the probability density function for the 4-parameter
' Generalized Gamma Distribution
' The "on error" is needed to integrate
Function lnGGD4pdf(X As Single, c As Single, a As Single, b As Single, p As Single) As Single
   Dim d As Single

   d = b * p
On Error GoTo err:
   lnGGD4pdf = Log(p) + (d - 1) * Log(X - c) - ((X - c) / a) ^ p - d * Log(a) - d * lnGamma(b)

   Exit Function
err:
   lnGGD4pdf = -200
End Function



'
' This is the function to be integrated.
' Set up for incomplete Gamma.
Function f(t As Double, b As Single) As Double
   If t <> 0 Then
      f = Exp(-t) * t ^ (b - 1)
   Else
      f = 0
   End If
End Function
```

```
'
' This Integrates f(x,b) from xFirst to xLast.
' Shammas Mathematical Algorithms in VB, pg 89
' McGraw-Hill, 1996
Function GaussLegendreQuadrature(xFirst As Single, xLast As Double, nSubIntervals As Integer, B1 As
Single) As Single

  Dim xA As Single, xB As Single
  Dim h As Single, hDiv2 As Single
  Dim Sum As Double, area As Double, xJ As Double
  Static Xk(5) As Single
  Static Ak(5) As Single
  Dim n As Integer, i As Integer, j As Integer

  Xk(0) = -0.9324695142
  Xk(1) = -0.6612093865
  Xk(2) = -0.2386191861
  Xk(3) = 0.2386191861
  Xk(4) = 0.6612093865
  Xk(5) = 0.9324695142
  Ak(0) = 0.1713244924
  Ak(1) = 0.360761573
  Ak(2) = 0.4679139346
  Ak(3) = 0.4679139346
  Ak(4) = 0.360761573
  Ak(5) = 0.1713244924
  area = 0
  n = nSubIntervals

  h = (xLast - xFirst) / n
  xA = xFirst
  For i = 1 To n
    Sum = 0
    xB = xA + h
    hDiv2 = h / 2
    ' obtain area of sub-interval
    For j = 0 To 5
       xJ = xA + hDiv2 * (Xk(j) + 1)
       Sum = Sum + Ak(j) * f(xJ, B1)
    Next j
    area = area + hDiv2 * Sum
    xA = xB
  Next i

  GaussLegendreQuadrature = area
End Function
```

```
' lnGamma function
' It returns the value of the LN(Gamma(XX) for XX>0
' Full accuracy is obtained for XX>1
'
' Numerical Recipes 11/15/92
Function lnGamma(xx As Single) As Single

    Dim i          As Integer

    Dim cof(6)     As Single
    Dim stp        As Single
    Dim X          As Single
    Dim tmp        As Single
    Dim ser        As Single


    cof(1) = 76.18009173
    cof(2) = -86.50532033
    cof(3) = 24.01409822
    cof(4) = -1.231739516
    cof(5) = 0.00120858003
    cof(6) = -0.00000536382

    stp = 2.5066282746

    X = xx - 1#
    tmp = X + 5.5
    tmp = (X + 0.5) * Log(tmp) - tmp

    ser = 1#

    For i = 1 To 6
      X = X + 1#
      ser = ser + cof(i) / X
    Next i

    lnGamma = tmp + Log(stp * ser)
End Function
```

```
'
'   ALGORITHM AS 103  APPL. STATIST. (1976) VOL.25, NO.3
'
'   Calculates DIGAMMA(X) = D( LOG( GAMMA(X))) / DX
'     Also known as the Psi function
Function DiGamma(X As Single) As Single

   Dim S    As Single
   Dim c    As Single
   Dim S3  As Single
   Dim S4  As Single
   Dim S5  As Single
   Dim d1  As Single
   Dim Y   As Single
   Dim R   As Single
'
'   Set constants, SN = Nth Stirling coefficient, D1 = DIGAMMA(1.0)
'

   Const ZERO = 0#
   Const HALF = 0.5
   Const ONE = 1#

   S = 0.00001
   c = 8.5
   S3 = 0.08333333333
   S4 = 0.0083333333333
   S5 = 0.003968253968
   d1 = -0.5772156649

   '
   '   Check argument is positive
   '
   DiGamma = ZERO
   Y = X

   '
   '   Use approximation if argument <= S
   '
   If (Y <= S) Then
      DiGamma = d1 - ONE / Y
      Return
   End If
   '
   '   Reduce to DiGamma(X + N) where (X + N) >= C
   '
line1:
   If (Y >= c) Then GoTo line2:
      DiGamma = DiGamma - ONE / Y
      Y = Y + ONE
   GoTo line1:
   '
```

```
    '   Use Stirling's (actually de Moivre's) expansion if argument > C
    '
line2:
    R = ONE / Y
    DiGamma = DiGamma + Log(Y) - HALF * R
    R = R * R
    DiGamma = DiGamma - R * (S3 - R * (S4 - R * S5))

End Function
```

MicroGA Module

```
Option Explicit
' Contains the Micro Genetic Algorithm Driver.

'
' This driver executes the Micro-GA algorithm and calculates an
' MLE & MDE estimate.
Sub MicroGA(TC1 As Single, TA1 As Single, TB1 As Single, TP1 As Single, TC2 As Single, TA2 As
Single, TB2 As Single, TP2 As Single, TM As Single, TrueParmUppLim As Single)

  Dim OldPop          As Population      ' Two non-overlapping populations
  Dim NewPop          As Population

  Dim BestIndividual  As IndividualRecord

  Dim n               As Integer          ' Number of variates
  Dim i       As Integer      '1=MLE, 2=MDE
  Dim Gen             As Integer

  Dim SumFitness      As Double
  Dim Avg             As Double
  Dim Max             As Double
  Dim Min             As Double

  Dim CheckFitness    As Double

  Dim X(500)          As Single       'Variates array

  Dim StartTime       As Date


  Sheets("output").Select
  Call HeaderBest

  n = pvMaxVariates


  Call GenerateRV(TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM, X, n) 'in RNG Mod
  Call SetupForMLE

 For i = 1 To 2

  StartTime = Time
  Gen = 0
  CheckFitness = -1.79769313486232E+302      ' A very negative number

  Call Initialize(OldPop, X, n, SumFitness, Max, Avg, Min, BestIndividual) 'in Init mod

  Do
```

```
      Gen = Gen + 1

      Call Generation(OldPop, NewPop, SumFitness, X, n, BestIndividual)      'in generation mod
      Call Statistics(NewPop, SumFitness, Max, Avg, Min, BestIndividual, False)   ' in stat mod

      'Call Report(Gen, OldPop, NewPop, Max, Min, Avg, SumFitness)


      OldPop = NewPop   ' advance the generation

      If MicroGAConvergance(OldPop, BestIndividual) Then      ' in ops mod
        Call InitPop(OldPop, X, n, False, BestIndividual)  ' in init mod
      End If

      If (Gen Mod pvCheck) = 0 Then
        If CheckFitness = BestIndividual.Fitness Then
          Exit Do
        Else
          CheckFitness = BestIndividual.Fitness
        End If
      End If
    Loop

    Call ProcessBestIndividual(BestIndividual, X, n, Gen, StartTime, _
        TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM, TrueParmUppLim)

  Next i

  End Sub
```

```
'
' This prints out the best individual,  and sets up variables for
' Minimum Distance and performs min dist and distance from true
Sub ProcessBestIndividual(BestIndividual As IndividualRecord, X() As Single, n As Integer, Gen As
Integer, StartTime As Date, _
   TC1 As Single, TA1 As Single, TB1 As Single, TP1 As Single, _
   TC2 As Single, TA2 As Single, TB2 As Single, TP2 As Single, _
   TM As Single, TrueParmUppLim As Single)

   Dim Distance      As Single
   Dim UpperLimit  As Single
   Dim NewM          As Single
   Dim NewC1         As Single
   Dim NewC2         As Single


With BestIndividual

   UpperLimit = FindUppLim(TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM, TrueParmUppLim) 'in int
dist mod

         'Integrate from lower limit c1 to Upperlimit
   Distance = IntegratedDistance(.C1, UpperLimit, 20, .C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M, TC1,
TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM)

   Call PrintBest(BestIndividual, n, Gen, StartTime, Distance)

   If Not pvMDE Then
      Call SetupForMDE
      Call RSort(X(), n)
      NewM = MinDistM(BestIndividual, X, n)      'Found in MinDistM Mod
      NewC1 = MinDistC1(BestIndividual, X, n)    'Found in MinDistC1 Mod

      .M = NewM
      .pvM = NewM

      .C1 = NewC1
      pvC1 = NewC1
      .Fitness = ObjFunc(.C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M, X, n)

   End If
End With
End Sub
```

```
'
' Prints the bestIndividual and all necessary statistic
' on the next line on the activesheet.
Sub PrintBest(Best As IndividualRecord, n As Integer, Gen As Integer, StartTime As Date, Dist As Single)
  Dim RowOut As Integer, Off As Integer
  Dim Better As String

With Best

  If pvMDE Then
    RowOut = Range("a1").CurrentRegion.Rows.Count
    Off = 1
  Else
    RowOut = Range("a1").CurrentRegion.Rows.Count + 1
    Off = 0
  End If

  Cells(RowOut, 1) = RowOut - 1
  Cells(RowOut, 3 + Off) = .C1 & " " & .A1 & " " & .B1 & " " & .P1 & ";" & .C2 & " " & .A2 & " " &
.B2 & " " & .P2 & ";" & .M

  Cells(RowOut, 5 + Off) = Dist
  Cells(RowOut, 7 + Off) = .Fitness
  Cells(RowOut, 9 + Off) = Format(StartTime - Time, "n,ss")
  Cells(RowOut, 11 + Off) = Gen


  If pvMDE Then

    If Dist < Cells(RowOut, 5).Value Then
      Better = "MDE"
    Else
      Better = "MLE"
    End If
  End If

  Cells(RowOut, 2) = Better

End With
End Sub
```

```
'
' Header for the Best individual report
'
Sub HeaderBest()

    Range("B1") = "Better"
    Range("C1") = "MLE Param"
    Range("D1") = "MDE Param"
    Range("E1") = "MLE Dist"
    Range("F1") = "MDE Dist"
    Range("G1") = "MLE Fit"
    Range("H1") = "MDE Fit"
    Range("I1") = "MLE Time"
    Range("J1") = "MDE Time"
    Range("K1") = "MLE Gen"
    Range("L1") = "MDE Gen"
End Sub
```

Interface Module

```
Option Explicit
' Contains the Objective Function and Decode routines for the GA.

'' This is the Fitness function
' It calculates the Sum of the logs of GGD9
' and penalizes for probilities = 0 and c2 > greatest variate.
Function ObjFunc(C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2 As Single,
    B2 As Single, P2 As Single, M As Single, X() As Single, n As Integer) As Double

  Dim i              As Integer

  Dim lnGGD4c1       As Double     'log of GGD4 Component1 pdf
  Dim lnGGD4c2       As Double     'log of GGD4 Component2 pdf
  Dim lnGGD9         As Double     'log of GGD9 pdf
  Dim GGD9           As Double     '    GGD9 pdf

  Dim Const1         As Single     'constant portion of GGD4 component1
  Dim Const2         As Single     'constant portion of GGD4 component2
  Dim D1minus1       As Single
  Dim D2minus1       As Single

  lnGGD9 = 0
  Const1 = Log(P1) - B1 * P1 * Log(A1) - lnGamma(B1)
  Const2 = Log(P2) - B2 * P2 * Log(A2) - lnGamma(B2)

  D1minus1 = B1 * P1 - 1
  D2minus1 = B2 * P2 - 1

  For i = 1 To n
    If X(i) > C2 Then
      lnGGD4c2 = Const2 + D2minus1 * Log(X(i) - C2) - ((X(i) - C2) / A2) ^ P2
    End If

    lnGGD4c1 = Const1 + D1minus1 * Log(X(i) - C1) - ((X(i) - C1) / A1) ^ P1

    GGD9 = M * Exp(lnGGD4c1) + (1 - M) * Exp(lnGGD4c2)

    If GGD9 > 0 Then
      lnGGD9 = lnGGD9 + Log(GGD9)
    Else
      lnGGD9 = lnGGD9 - 50      'Penalty
    End If
  Next i

  If C2 >= pvBiggestVariate Then lnGGD9 = lnGGD9 - 200

  ObjFunc = lnGGD9
End Function
```

```
'
' Decode string as unsigned binary integer - true=1, false=0
' C2 always > c1.
Sub Decode(Chrom() As Boolean, C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single,
A2 As Single, B2 As Single, P2 As Single, M As Single)
  Dim i As Integer

  Dim C2Tot As Single
  Dim A1Tot As Single
  Dim A2Tot As Single
  Dim B1Tot As Single
  Dim B2Tot As Single
  Dim P1Tot As Single
  Dim P2Tot As Single
  Dim MTot  As Single
  Dim PowerOf2 As Single

  C2Tot = pvSmallestParam              'to avoid zero's
  A1Tot = pvSmallestParam
  A2Tot = pvSmallestParam
  B1Tot = pvSmallestParam
  B2Tot = pvSmallestParam
  P1Tot = pvSmallestParam
  P2Tot = pvSmallestParam
  MTot = pvSmallestParam

  PowerOf2 = pvSmallestParam
  For i = 1 To 10
    PowerOf2 = PowerOf2 * 2
    If Chrom(i) Then A1Tot = A1Tot + PowerOf2
    If Chrom(i + 10) Then A2Tot = A2Tot + PowerOf2
    If Chrom(i + 20) Then P1Tot = P1Tot + PowerOf2
    If Chrom(i + 30) Then P2Tot = P2Tot + PowerOf2
    If Chrom(i + 40) Then B1Tot = B1Tot + PowerOf2
    If Chrom(i + 50) Then B2Tot = B2Tot + PowerOf2
  '  If Chrom(i + 60) Then C2Tot = C2Tot + PowerOf2
  Next i


  If Not pvMDE Then

    PowerOf2 = pvSmallestParam
    For i = 1 To 10
      PowerOf2 = PowerOf2 * 2
      If Chrom(i + 60) Then C2Tot = C2Tot + PowerOf2
    Next i

    PowerOf2 = pvSmallestParam
    For i = 1 To 6
      PowerOf2 = PowerOf2 * 2
      If Chrom(i + 70) Then MTot = MTot + PowerOf2
```

```
    Next i

    M = MTot
    C2 = C2Tot + pvC1
  Else
    M = pvM
    C2 = pvC2
  End If

  C1 = pvC1
  A1 = A1Tot
  B1 = B1Tot
  P1 = P1Tot

  A2 = A2Tot
  B2 = B2Tot
  P2 = P2Tot
End Sub
```

```
'
' Calculates the summations needed for the log likelihood function
Sub LikelihoodSums(C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2 As
Single, B2 As Single, P2 As Single, M As Single, _
 X() As Single, n As Integer, SumLNu As Single, SumUtoP As Single, SumUtoPlnU As Single)

  Dim i          As Integer

  Dim U          As Double
  Dim UtoP       As Double
  Dim lnU        As Double


  SumUtoP = 0
  SumLNu = 0

   For i = 1 To n
     U = X(i) - C1
     UtoP = U ^ P1
     lnU = Log(U)

     SumUtoP = SumUtoP + UtoP
     SumLNu = SumLNu + lnU
     SumUtoPlnU = SumUtoPlnU + UtoP * lnU
   Next i
End Sub


' 3rd Equation of Parr & Webster
' A method for Discriminating Between Failure Density Functions Used
'        in Reliability Predictions
' Technometrics  Vol 7, No 1, pg 1-10 (Feb 1963)
' Used to find 0 of GGD4
Function Parr3(n As Integer, p As Single, b As Single, SumLNu As Single, SumUtoP As Single) As Double
   Dim d As Double
   d = b * p
   Parr3 = Log(p) - Log(n) - Log(d) + Log(SumUtoP) + DiGamma(d / p) - (p / n) * SumLNu
End Function
```

```
' 4rd Equation of Parr & Webster
' A method for Discriminating Between Failure Density Functions Used
'       in Reliability Predictions
' Technometrics  Vol 7, No 1, pg 1-10 (Feb 1963)
' Used to find 0 of GGD4
Function Parr4(n As Integer, p As Single, b As Single, SumUtoP As Single, SumUtoPlnU As Single) As
Double
   Dim d As Double
   d = b * p
   Parr4 = Log(p) - Log(n) - Log(d) + Log(SumUtoP) + DiGamma(d / p) + (p / d) - (p / SumUtoP) *
SumUtoPlnU
End Function
```

Generation Module

```
Option Explicit
' Contains the GA routines that create a new generation.

' Create a new generation through select, crossover, and mutation
Sub Generation(OldPop As Population, NewPop As Population, SumFitness As Double, X() As Single, n
As Integer, BestInd As IndividualRecord)

  Dim Cnt       As Integer
  Dim j As Integer
  Dim Mate1     As Integer
  Dim Mate2     As Integer
  Dim NextMate As Integer
  Dim jCross    As Integer

  NextMate = pvPopSize

  For j = 1 To 3 Step 2  ' select, crossover, and mutation loop until newpop is filled

    Call SelectMate(Mate1, NextMate, OldPop)
    Call SelectMate(Mate2, NextMate, OldPop)

' Crossover
    Call CrossOver(OldPop.Individual(Mate1).Chrom, OldPop.Individual(Mate2).Chrom, _
      NewPop.Individual(j).Chrom, NewPop.Individual(j + 1).Chrom, _
      pvLChrom, jCross)

    ' Decode string, evaluate fitness, & record parentage date on both children

    With NewPop.Individual(j)

       Call Decode(.Chrom, .C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M)
      .Fitness = ObjFunc(.C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M, X, n)

      .Parent1 = Mate1
      .Parent2 = Mate2
      .Xsite = jCross
    End With

    With NewPop.Individual(j + 1)

       Call Decode(.Chrom, .C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M)
      .Fitness = ObjFunc(.C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M, X, n)
      .Parent1 = Mate1
      .Parent2 = Mate2
      .Xsite = jCross
    End With
    ' Increment population index
  Next j
```

```
  NewPop.Individual(pvPopSize) = BestInd
End Sub
```

Operations Module

```
Option Explicit
' Contains the GA operators

' 3-operators: Reproduction (selectMate), Crossover (crossover),
'           & Mutation (mutation)


'
' Selects a mate and then advances the MatePointer
' Randomly sorts the population and resets the MatePointer
' as needed.
Sub SelectMate(Mate As Integer, MatePointer As Integer, OldPop As Population)

   If MatePointer >= pvPopSize Then
      Call RandomSortPopulation(OldPop)
      MatePointer = 1
   End If

   If OldPop.Individual(MatePointer).Fitness > OldPop.Individual(MatePointer + 1).Fitness Then
      Mate = MatePointer
   Else
      Mate = MatePointer + 1
   End If

   MatePointer = MatePointer + 2
End Sub

' Select a single individual via roulette wheel selection
Function SelectInd(Pop As Population, SumFitness As Double) As Integer

   Dim Rand       As Double
   Dim PartSum    As Double   ' Random point on wheel, partial sum

   Dim j As Integer           ' population index

   PartSum = 0#
   j = 0       ' Zero out counter and accumulator
   Rand = rnd * SumFitness   ' Wheel point calc. uses random number (0,1)
   Do   ' Find wheel slot
      j = j + 1
      PartSum = PartSum + Pop.Individual(j).Fitness
   Loop Until (PartSum >= Rand) Or (j = pvPopSize)

   SelectInd = j       ' Return individual number
End Function
```

```
' Cross 2 parent strings, place in 2 child strings
Sub CrossOver(Parent1, Parent2, child1, child2, pvLChrom, jCross As Integer)
  Dim i As Integer
  Dim j As Integer

    jCross = RandomInteger(1, pvLChrom - 1)   ' Cross between 1 and l-1
    pvNCross = pvNCross + 1          ' Increment crossover counter


 ' 1st exchange, 1 to 1 and 2 to 2
  For i = 1 To pvLChrom
    child1(i) = Parent1(i)
    child2(i) = Parent2(i)
  Next i

 ' 2nd exchange, 1 to 2 and 2 to 1
    For j = jCross + 1 To pvLChrom
      child1(j) = Parent2(j)
      child2(j) = Parent1(j)
    Next j
End Sub

' Test to see if the Convergence rule has been met.
' from Carroll's GAfortran 1.6.4 subroutine gamicro.
Function MicroGAConvergance(OldPop As Population, BestInd As IndividualRecord) As Boolean

  Dim i As Integer
  Dim j As Integer
  Dim Count     As Integer

  Count = 0
  For i = 1 To pvPopSize
    For j = 1 To pvLChrom
      If Not BestInd.Chrom(j) = OldPop.Individual(i).Chrom(j) Then
        Count = Count + 1
      End If
    Next j
  Next i

  If Count < 0.05 * (pvPopSize - 1) * pvLChrom Then
    MicroGAConvergance = True
  Else
    MicroGAConvergance = False
  End If

 End Function
```

```
'
' Sorts the population into a random order
'
Sub RandomSortPopulation(Pop As Population)

  Dim i  As Integer
  Dim SwapTo   As Integer

  Dim Hold        As IndividualRecord


  For i = 1 To pvPopSize
    Hold = Pop.Individual(i)

    SwapTo = RandomInteger(1, pvPopSize)    'Found in Rand mod

    Pop.Individual(i) = Pop.Individual(SwapTo)
    Pop.Individual(SwapTo) = Hold
  Next i

End Sub
```

Rand Module

```
Option Explicit
' Contains the random number operators needed for the GA.


' Flip a biased coin - true if heads
Function Flip(probability As Double) As Boolean
  Flip = (rnd <= probability)
End Function


' Pick a random integer between low and high
Function RandomInteger(Low As Integer, High As Integer) As Integer
  Dim i As Integer

  If Low >= High Then
    i = Low
  Else
    i = Fix(rnd * (High - Low + 1) + Low)   'return an integer
    If i > High Then i = High
  End If

  RandomInteger = i
End Function
```

Initial Module

```
Option Explicit
'Contains the initialization routines for the GA

'
' Initialization Driver
'
Sub Initialize(OldPop As Population, X() As Single, n As Integer, SumFitness As Double, Max1 As
Double, Avg1 As Double, Min1 As Double, BestInd As IndividualRecord)
   Randomize

   Call InitPop(OldPop, X, n, True, BestInd)
   Call Statistics(OldPop, SumFitness, Max1, Avg1, Min1, BestInd, True)
End Sub

'
' Initialize a population at random
' if first call initialize the whole population, otherwise
'     initialize n-1 and keep the best individual
Sub InitPop(Pop As Population, X() As Single, n As Integer, Initial As Boolean, BestInd As
IndividualRecord)
   Dim i         As Integer
   Dim j         As Integer
   Dim NumNew As Integer

   If Initial And Not pvMDE Then
      NumNew = pvPopSize
      Call VariateStatistics(X, n)
      pvC1 = pvMaxLocParam
   Else
      NumNew = pvPopSize - 1
      Pop.Individual(pvPopSize) = BestInd
   End If

   For i = 1 To NumNew
      With Pop.Individual(i)
         For j = 1 To pvLChrom
            .Chrom(j) = Flip(0.5)   ' A fair coin toss
         Next j

         Call Decode(.Chrom, .C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M) 'in inteface Mod
         .Fitness = ObjFunc(.C1, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M, X, n)       'in inteface Mod
         .Parent1 = 0
         .Parent2 = 0
         .Xsite = 0
      End With
   Next i
End Sub
'
```

```
' This finds the Smallest Variate and then sets the
' c1 location parameter based on its value.
' and the biggest variate.
Sub VariateStatistics(X() As Single, n As Integer)

   Dim i            As Integer

   Dim SmallestX        As Single
   Dim BiggestX         As Single

   SmallestX = 3.402823E+38    'Largest Value stored in single precision
   BiggestX = -3E+38

   For i = 1 To n
      If X(i) < SmallestX Then
         SmallestX = X(i)
      End If

      If X(i) > BiggestX Then
         BiggestX = X(i)
      End If

   Next i

   If SmallestX - pvSmallestParam < 0 Then
      pvMaxLocParam = 0
   Else
      pvMaxLocParam = SmallestX - pvSmallestParam
   End If

   pvBiggestVariate = BiggestX

End Sub

' Not currently used.
 ' Interactive data inquiry and setup
Sub InitData()

   Call Output("--------------------------------")
   Call Output("A Simple Genetic Algorithm - SGA")
   Call Output(" (c) David Edward Goldberg 1986")
   Call Output("    All Rights Reserved       ")
   Call Output("--------------------------------")

   Call Output("******** SGA Data Entry and Initialization ************")
   Call Output
   Call Output("population size ------- > " & pvPopSize)
   Call Output("chromosome length ----- > " & pvLChrom)
   Call Output("   Check every X generations (pvCheck)  =   " & pvCheck)

End Sub
```

```
' Not currently Used
' Initial report
Sub InitReport(SumFitness, Max, Avg, Min)

 Call Output("--------------------------------------------------")
 Call Output("|   A Simple Genetic Algorithm - SGA - v1.0    |")
 Call Output("|   (c)   David Edward Goldberg 1986         |")
 Call Output("|           All Rights Reserved            |")
 Call Output("--------------------------------------------------")

 Call Output("    SGA Parameters")
 Call Output("    --------------")

 Call Output("  Population size (pvPopSize)        =  " & pvPopSize)
 Call Output("  Chromosome length (pvLChrom)        =  " & pvLChrom)
 Call Output("  Check every X generations (pvCheck) =  " & pvCheck)

 Call Output("    Initial Generation Statistics")
 Call Output("    ----------------------------")

 Call Output("  Initial population maximum fitness = " & Max)
 Call Output("  Initial population average fitness = " & Avg)
 Call Output("  Initial population minimum fitness = " & Min)
 Call Output("  Initial population sum of fitness  = " & SumFitness)

End Sub
```

Report Module

```
Option Explicit
' Contains the test report for the Ga.
' Only used for testing.


'
' This prints out all the individuals for the population
'
Sub Report(Gen As Integer, OldPop As Population, NewPop As Population, Max, Min, Avg, Sum)

   Dim i            As Integer
   Dim RowOut            As Integer

   Dim ChromoString      As String

   Worksheets("output").Select
   RowOut = Range("a1").CurrentRegion.Rows.Count + 1

   Cells(RowOut, 1) = "Generations" & Gen - 1 & "," & Gen

   For i = 1 To pvPopSize
      RowOut = Range("a1").CurrentRegion.Rows.Count + 1

      Cells(RowOut, 1) = i
      With OldPop.Individual(i)
         Call ChromToString(ChromoString, .Chrom)
         Cells(RowOut, 2) = ChromoString
         Cells(RowOut, 3) = .C1 & " " & .A1 & " " & .B1 & " " & .P1 & ";" & .C2 & " " & .A2 & " " & .B2
& " " & .P2 & ";" & .M
         Cells(RowOut, 4) = .Fitness
      End With

      With NewPop.Individual(i)
         Cells(RowOut, 5) = .Parent1 & "," & .Parent2
         Cells(RowOut, 6) = .Xsite
         Call ChromToString(ChromoString, .Chrom)
         Cells(RowOut, 7) = ChromoString
         Cells(RowOut, 8) = .C1 & " " & .A1 & " " & .B1 & " " & .P1 & ";" & .C2 & " " & .A2 & " " & .B2
& " " & .P2 & ";" & .M
         Cells(RowOut, 9) = .Fitness
      End With
   Next i

   RowOut = Range("a1").CurrentRegion.Rows.Count + 1
      Cells(RowOut, 1) = "Max"
      Cells(RowOut, 2) = "Min"
      Cells(RowOut, 3) = "average"
      Cells(RowOut, 4) = "SumFitness"
      Cells(RowOut, 5) = "Mutates"
```

```
    Cells(RowOut, 6) = "Cross"

    Cells(RowOut + 1, 1) = Max
    Cells(RowOut + 1, 2) = Min
    Cells(RowOut + 1, 3) = Avg
    Cells(RowOut + 1, 4) = Sum
    Cells(RowOut + 1, 5) = pvNumMutation
    Cells(RowOut + 1, 6) = pvNCross

    Cells(RowOut + 2, 1) = String(Number:=100, Character:="*")
End Sub

'
' Header Macro
' Macro recorded 11/14/97 by Dean Boerrigter'
Sub Header()
  Range("a1").FormulaR1C1 = "#"
  Range("B1").FormulaR1C1 = "String"
  Range("C1").FormulaR1C1 = "X"
  Range("D1").FormulaR1C1 = "Fitness"
  Range("E1").FormulaR1C1 = "Parents"
  Range("F1").FormulaR1C1 = "Xsite"
  Range("G1").FormulaR1C1 = "String"
  Range("H1").FormulaR1C1 = "x"
  Range("I1").FormulaR1C1 = "Fitness"

  Range("A2").Select
  ActiveWindow.FreezePanes = True
End Sub

' this write the String to the Output worksheet on the
' next empty line
Sub Output(Optional StringOut)
  Dim RowOut As Integer

  With Sheets("output")
    RowOut = .Range("a1").CurrentRegion.Rows.Count + 1
    .Cells(RowOut, 1) = StringOut
  End With
End Sub

' write a chromosome as a string of 1"s (true"s) and 0"s (false"s)
Sub ChromToString(TextOut As String, Chrom)
Dim j As Integer
  TextOut = ""
  For j = pvLChrom To 1 Step -1
    If Chrom(j) Then
      TextOut = TextOut + "1"
    Else
      TextOut = TextOut + "0"
    End If
  Next j
End Sub
```

Stat Module

```
Option Explicit
' Calculates the statistics for the GA.

 ' Calculate population statistics
Sub Statistics(Pop As Population, SumFitness As Double, Maximum As Double, Avg1 As Double,
Minimum As Double, TopIndividual As IndividualRecord, Initialize As Boolean)

  Dim j As Integer


  If Initialize Then
    Minimum = Pop.Individual(1).Fitness
    Maximum = Pop.Individual(1).Fitness
    TopIndividual.Fitness = -1.79769313486232E+302
  End If

  SumFitness = 0
  For j = 1 To pvPopSize
    With Pop.Individual(j)
      SumFitness = SumFitness + .Fitness  ' Accumulate fitness sum
      If .Fitness >= Maximum Then Maximum = .Fitness            ' New Current Maximum

      If .Fitness >= TopIndividual.Fitness Then
        TopIndividual = Pop.Individual(j)  ' Save Best Individual
      End If

      If .Fitness < Minimum Then Minimum = .Fitness ' New Current Minimum
    End With
  Next j

  Avg1 = SumFitness / pvPopSize
End Sub
```

MinDistM Module

```
Option Explicit
'Contains the Minimum Distance Routines for the mixture parameter, m.


' finds the Minimum Distance for the mixture parameter
Function MinDistM(BestInd As IndividualRecord, X() As Single, n As Integer) As Single
With BestInd

  Call RSort(X(), n)  'Found in Sort Mod
  MinDistM = GoldenSearchMinM(pvSmallestParam, 1 - pvSmallestParam, 0.0001, X, n, .C1, .A1, .B1,
.P1, .C2, .A2, .B2, .P2)

End With
End Function

' Finds the minimum value within an interval.
' Mathematical Algorithms in VB for Scientist & Engineers
' Shammas, Nammar, 1996.  pg 115-116
Function GoldenSearchMinM(xA As Double, xB As Double, tolerance As Double, Variates() As Single, n
As Integer, C1 As Single, A1 As Single, B1 As Single, P1 As Single, _
   C2 As Single, A2 As Single, B2 As Single, P2 As Single) As Single

 Const MaxIter = 1000
 Dim Xc As Single, Xd As Single
 Dim Fc As Single, Fd As Single
 Dim oneMinusTau As Single
 Dim iter As Integer

 iter = 0
 oneMinusTau = 1 - (Sqr(5) - 1) / 2
 Xc = xA + oneMinusTau * (xB - xA)
 Fc = CalcAD(Variates, n, C1, A1, B1, P1, C2, A2, B2, P2, Xc)
 Xd = xB - oneMinusTau * (xB - xA)
 Fd = CalcAD(Variates, n, C1, A1, B1, P1, C2, A2, B2, P2, Xd)
 Do
  iter = iter + 1
  If Fc < Fd Then
   xB = Xd
   Xd = Xc
   Xc = xA + oneMinusTau * (xB - xA)
   Fd = Fc
   Fc = CalcAD(Variates, n, C1, A1, B1, P1, C2, A2, B2, P2, Xc)
  Else
   xA = Xc
   Xc = Xd
   Xd = xB - oneMinusTau * (xB - xA)
   Fc = Fd
   Fd = CalcAD(Variates, n, C1, A1, B1, P1, C2, A2, B2, P2, Xd)
  End If
```

```
  Loop While Abs(xB - xA) > tolerance And iter < MaxIter
  If iter <= MaxIter Then
    GoldenSearchMinM = Xc
  Else
    GoldenSearchMinM = -31
  End If
End Function
```

MinDistC1 Module

```
Option Explicit
'Contains the Minimum Distance Routines for the C1 parameter.



' finds the Minimum Distance for the C1 parameter
Function MinDistC1(BestInd As IndividualRecord, X() As Single, n As Integer) As Single
  Dim MaxC1 As Single
With BestInd
  MaxC1 = pvMaxLocParam

  MinDistC1 = GoldenSearchMinC1(0, MaxC1, 0.0001, X, n, .A1, .B1, .P1, .C2, .A2, .B2, .P2, .M)

End With
End Function


' Finds the minimum value within an interval.
' Mathematical Algorithms in VB for Scientist & Engineers
' Shammas, Nammar, 1996. pg 115-116
Function GoldenSearchMinC1(xA As Single, xB As Single, tolerance As Double, Variates() As Single, n
As Integer, A1 As Single, B1 As Single, P1 As Single, _
   C2 As Single, A2 As Single, B2 As Single, P2 As Single, M As Single) As Single

 Const MaxIter = 1000
 Dim Xc As Single, Xd As Single
 Dim Fc As Single, Fd As Single
 Dim oneMinusTau As Single
 Dim iter As Integer

 iter = 0
 oneMinusTau = 1 - (Sqr(5) - 1) / 2
 Xc = xA + oneMinusTau * (xB - xA)
 Fc = CalcAD(Variates, n, Xc, A1, B1, P1, C2, A2, B2, P2, M)
 Xd = xB - oneMinusTau * (xB - xA)
 Fd = CalcAD(Variates, n, Xd, A1, B1, P1, C2, A2, B2, P2, M)

 Do
  iter = iter + 1
  If Fc < Fd Then
    xB = Xd
    Xd = Xc
    Xc = xA + oneMinusTau * (xB - xA)
    Fd = Fc
    Fc = CalcAD(Variates, n, Xc, A1, B1, P1, C2, A2, B2, P2, M)
  Else
    xA = Xc
    Xc = Xd
    Xd = xB - oneMinusTau * (xB - xA)
```

```
    Fc = Fd
    Fd = CalcAD(Variates, n, Xd, A1, B1, P1, C2, A2, B2, P2, M)
  End If
 Loop While Abs(xB - xA) > tolerance And iter < MaxIter
 If iter <= MaxIter Then
  GoldenSearchMinC1 = Xc
 Else
  GoldenSearchMinC1 = -31
 End If
End Function
```

MinDistC2 Module

```
Option Explicit

'
' finds the Minimum Distance for the C2 parameter
'
Function MinDistC2(BestInd As IndividualRecord, X() As Single, n As Integer) As Single
   Dim MaxC1 As Single

With BestInd

   MaxC1 = pvBiggestVariate - pvSmallestParam

   MinDistC2 = GoldenSearchMinC2(pvC1 + pvSmallestParam, MaxC1, 0.0001, X, n, .C1, .A1, .B1, .P1,
.A2, .B2, .P2, .M)

End With
End Function




' Finds the minimum value within an interval.
' Mathematical Algorithms in VB for Scientist & Engineers
' Shammas, Nammar, 1996.  pg 115-116
Function GoldenSearchMinC2(xA As Single, xB As Single, tolerance As Double, Variates() As Single, n
As Integer, C1 As Single, A1 As Single, B1 As Single, P1 As Single, _
    A2 As Single, B2 As Single, P2 As Single, M As Single) As Single

  Const MaxIter = 1000
  Dim Xc As Single, Xd As Single
  Dim Fc As Single, Fd As Single
  Dim oneMinusTau As Single
  Dim iter As Integer

  iter = 0
  oneMinusTau = 1 - (Sqr(5) - 1) / 2
  Xc = xA + oneMinusTau * (xB - xA)
  Fc = CalcAD(Variates, n, C1, A1, B1, P1, Xc, A2, B2, P2, M)
  Xd = xB - oneMinusTau * (xB - xA)
  Fd = CalcAD(Variates, n, C1, A1, B1, P1, Xd, A2, B2, P2, M)

  Do
    iter = iter + 1
    If Fc < Fd Then
      xB = Xd
      Xd = Xc
      Xc = xA + oneMinusTau * (xB - xA)
      Fd = Fc
      Fc = CalcAD(Variates, n, C1, A1, B1, P1, Xc, A2, B2, P2, M)
```

```
    Else
      xA = Xc
      Xc = Xd
      Xd = xB - oneMinusTau * (xB - xA)
      Fc = Fd
      Fd = CalcAD(Variates, n, C1, A1, B1, P1, Xd, A2, B2, P2, M)
    End If
  Loop While Abs(xB - xA) > tolerance And iter < MaxIter
  If iter <= MaxIter Then
    GoldenSearchMinC2 = Xc
  Else
    GoldenSearchMinC2 = -31
  End If
End Function
```

AD Module

```
Option Explicit
' Contains the Anderson Darling Test routines.

'
' Converts the variates to their CDF probability
' and then calculates the Anderson-Darling Statistic
Function CalcAD(X() As Single, n As Integer, C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2
As Single, A2 As Single, B2 As Single, P2 As Single, M As Single) As Single

   Dim Z(500) As Single


   Call XtoZ(X, Z, n, C1, A1, B1, P1, C2, A2, B2, P2, M)

   CalcAD = AndersonDarling(Z, n)
End Function

'
' Translates the variates to their cdf values.
'
Sub XtoZ(X() As Single, Z() As Single, n As Integer, C1 As Single, A1 As Single, B1 As Single, P1 As
Single, C2 As Single, A2 As Single, B2 As Single, P2 As Single, M As Single)

   Dim i As Integer

   For i = 1 To n
      Z(i) = GGD9cdf(X(i), C1, A1, B1, P1, C2, A2, B2, P2, M)
      If Z(i) >= 1 Then
         Z(i) = 0.999999
      Else
         If Z(i) = 0 Then Z(i) = 0.000001
      End If
   Next i
End Sub

'
' Anderson Darling test statistic
' Stephens. "EDF Statistics" JASA Vol 69,No.347, Pg 731
Function AndersonDarling(Z() As Single, n As Integer) As Single
   Dim i As Integer
   Dim Sum      As Single

   For i = 1 To n
      Sum = Sum + (2 * i - 1) * (Log(Z(i)) + Log(1 - Z(n + 1 - i)))
   Next i

   AndersonDarling = -Sum / n - n
End Function
```

Sort Module

```
Option Explicit
' Contains routines for sorting an array into ascending order


' sorts an array into ascending order
' Mumford, pg. 140
Sub RSort(X() As Single, n As Integer)

  Dim i        As Integer
  Dim j        As Integer
  Dim Low      As Integer

  For i = 1 To n - 1
    Low = i
    For j = i + 1 To n
      If X(j) < X(Low) Then
        Low = j
      End If
    Next j
    Call Swap(X(i), X(Low))
  Next i


End Sub

'
' Swaps two real values
Sub Swap(r1 As Single, r2 As Single)

  Dim temp As Single

  temp = r1
  r1 = r2
  r2 = temp
End Sub
```

RVG Module

```
Option Explicit
'Contains the Random Variate Generation routines.

'
' Generate the random variates.
'
Sub GenerateRV(C1 As Single, A1 As Single, B1 As Single, P1 As Single, _
        C2 As Single, A2 As Single, B2 As Single, P2 As Single, _
        M As Single, RV() As Single, n As Integer)

  Dim i As Integer

  For i = 1 To n
    RV(i) = GGD9RVG(C1, A1, B1, P1, C2, A2, B2, P2, M)
  Next i

End Sub

'
' Returns a 9-parameter Mixed Generalized Gamma Variate
' m = mixing proportion
'
Function GGD9RVG(C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2 As
Single, B2 As Single, P2 As Single, M As Single) As Single
  If rnd <= M Then
    GGD9RVG = GGD4RVG(C1, A1, B1, P1)
  Else
    GGD9RVG = GGD4RVG(C2, A2, B2, P2)
  End If
End Function

' Returns a 4-parameter Generalized Gamma Dist variate
'           Location  scale   power/shape power
'  Harter     c        a         b          p
'  for b>.25
'  Transforms from Tadikamila "Computing", (1979)
Function GGD4RVG(c As Single, a As Single, b As Single, p As Single) As Single

  Dim Z         As Single
  Dim X         As Single
  Dim Y         As Single

  Z = GBH(b)
  X = Z ^ (1 / p)

  GGD4RVG = X * a + c
End Function
```

```
' Cheng & Feast (Comm of the ACM, 1980)
' For Alp > 0.25
' Found in Tadikamalla & Johnston (1980)
'   Amer J. of Math. & Manage Sci. "A complete guide to
'   Gamma Variate Generation"
Function GBH(Alp As Single) As Single

   Dim a          As Single
   Dim b          As Single
   Dim c          As Single
   Dim d          As Single
   Dim t          As Single
   Dim h1         As Single
   Dim h2         As Single
   Dim U          As Single
   Dim U1         As Single
   Dim U2         As Single
   Dim w          As Single

a = Alp - 0.25
b = Alp / a
c = 2# / a
d = c + 2#
t = 1# / Sqr(Alp)
h1 = (0.4417 + 0.245 * t / Alp) * t
h2 = (0.222 - 0.043 * t) * t

line1:
   U1 = rnd
     U = rnd
   U2 = U1 + h1 * U - h2

   If (U2 <= 0) Then GoTo line1:
   If (U2 > 1) Then GoTo line1:
   w = b * (U1 / U2) ^ 4

   If w = 0 Then GoTo line1:   'added since Excel generates rng=0
   If c * U2 - d + w + 1 / w <= 0 Then GoTo line4:              'Goto line 4
   If c * Log(U2) - Log(w) + w - 1 >= 0 Then GoTo line1:

line4:
GBH = a * w
End Function
```

IntegratedDist Module

```
Option Explicit
' Contains the routines for calculation the Integrated Distance.



' This finds the upper limit of integration
' the point at which F(X)>0.999
' Xin is the point to start the search for the upper limit.
Function FindUppLim(C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2 As
Single, B2 As Single, P2 As Single, M As Single, Xin As Single) As Single

    Dim X As Single

    X = Xin

    While (GGD9cdf(X, C1, A1, B1, P1, C2, A2, B2, P2, M) < 0.999) And (X < 50)
        X = X + 0.05
    Wend

    FindUppLim = X
End Function


'
' This compares the pdf of the estimated and true parameters at
' point x and then squares the difference
Function Comp(X As Single, C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2
As Single, B2 As Single, P2 As Single, M As Single, _
                    TC1 As Single, TA1 As Single, TB1 As Single, TP1 As Single, TC2 As Single, TA2 As
Single, TB2 As Single, TP2 As Single, TM As Single) As Single

    Dim dif As Single

    dif = GGD9pdf(X, C1, A1, B1, P1, C2, A2, B2, P2, M) - GGD9pdf(X, TC1, TA1, TB1, TP1, TC2, TA2,
TB2, TP2, TM)

    Comp = dif ^ 2
End Function
```

```
' GaussLegendreQuadrature
' This Integrates f(x,b) from xFirst to xLast.
' Shammas Mathematical Algorithms in VB, pg 89
' McGraw-Hill, 1996
Function IntegratedDistance(xFirst As Single, xLast As Single, nSubIntervals As Integer, _
   C1 As Single, A1 As Single, B1 As Single, P1 As Single, C2 As Single, A2 As Single, B2 As Single, P2
As Single, M As Single, _
 TC1 As Single, TA1 As Single, TB1 As Single, TP1 As Single, TC2 As Single, TA2 As Single, TB2 As
Single, TP2 As Single, TM As Single) As Single

  Dim xA As Single, xB As Single, xJ As Single
  Dim h As Single, hDiv2 As Single
  Dim Sum As Single, area As Single
  Static Xk(5) As Single
  Static Ak(5) As Single
  Dim n As Integer, i As Integer, j As Integer

  Xk(0) = -0.9324695142
  Xk(1) = -0.6612093865
  Xk(2) = -0.2386191861
  Xk(3) = 0.2386191861
  Xk(4) = 0.6612093865
  Xk(5) = 0.9324695142
  Ak(0) = 0.1713244924
  Ak(1) = 0.360761573
  Ak(2) = 0.4679139346
  Ak(3) = 0.4679139346
  Ak(4) = 0.360761573
  Ak(5) = 0.1713244924
  area = 0
  n = nSubIntervals

  h = (xLast - xFirst) / n
  xA = xFirst
  For i = 1 To n
    Sum = 0
    xB = xA + h
    hDiv2 = h / 2
    ' obtain area of sub-interval
    For j = 0 To 5
      xJ = xA + hDiv2 * (Xk(j) + 1)
      Sum = Sum + Ak(j) * Comp(xJ, C1, A1, B1, P1, C2, A2, B2, P2, M, _
                  TC1, TA1, TB1, TP1, TC2, TA2, TB2, TP2, TM)
    Next j
    area = area + hDiv2 * Sum
    xA = xB
  Next i

  IntegratedDistance = area
End Function
```

Appendix C Summarization Code

Summary Mod

```
'
' Summarizes all the worksheets in the active book and
' checks for completeness
Sub Driver()
   Worksheets.Add before:=Sheets(1)
   ActiveSheet.Name = "Summary"
   Call SummaryHeader

   Call SummarizeAll
   Call setupPrintDriver
   Call Check

   ActiveWorkbook.Save
   Beep
End Sub

'
' Summarizes all the worksheets in the active book.
" "Summary Worksheet" must be in activebook.
Sub SummarizeAll()
Dim wsht As Object
   For Each wsht In ActiveWorkbook.Worksheets
      wsht.Select

      If Not wsht.Name = "Summary" Then
         Call Summarize
      End If

   Next wsht
End Sub

'
' Summarize the active sheet.
' "Summary Worksheet" must be in activebook.
Sub Summarize()

   Dim RowOut As Integer
   Dim NumRows As Integer
   Dim NumReplications As Integer

   Dim d As String

   'Count number of times MDE and MLE are better
   Range("Z1").FormulaR1C1 = "=COUNTIF(R2C2:R1001C2,""MDE"")"
   Range("Z2").FormulaR1C1 = "=COUNTIF(R2C2:R1001C2,""MLE"")"
```

```
With Sheets("summary")
   mleDistCol = GetCol("MLE Dist")
   mdeDistCol = GetCol("MDE Dist")
   MLETime = GetCol("MLE Time")
   MDETime = GetCol("MDE Time")

   RowOut = .Range("a1").CurrentRegion.Rows.Count + 1

   Call AverageSummary(MLEAveTime, MDEAveTime, NumReplications)
   d = ActiveSheet.Name

   .Cells(RowOut, 1) = Range("A1").NoteText
   .Cells(RowOut, 2).FormulaR1C1 = Range("z1").Value / (Range("z1").Value + Range("z2").Value)

   .Cells(RowOut, 3).FormulaR1C1 = "=AVERAGE(" & ActiveSheet.Name & "!R2C" & mleDistCol &
":R1001C" & mleDistCol & ")"
   .Cells(RowOut, 4).FormulaR1C1 = "=STDEV(" & ActiveSheet.Name & "!R2C" & mleDistCol &
":R1001C" & mleDistCol & ")"
   .Cells(RowOut, 5).FormulaR1C1 = "=AVERAGE(" & ActiveSheet.Name & "!R2C" & mdeDistCol &
":R1001C" & mdeDistCol & ")"
   .Cells(RowOut, 6).FormulaR1C1 = "=STDEV(" & ActiveSheet.Name & "!R2C" & mdeDistCol &
":R1001C" & mdeDistCol & ")"


   .Cells(RowOut, 7) = Format(MLEAveTime, "0.0")
   .Cells(RowOut, 8) = Format(MDEAveTime, "0.0")

   .Cells(RowOut, 9) = NumReplications
   .Cells(RowOut, 10) = d

   .Select
End With
   NumRows = Range("a1").CurrentRegion.Rows.Count
   Cells(NumRows + 1, 2).Select

End Sub

'
' This calculates some statistics for the summary
'
Sub AverageSummary(MLEAveTime, MDEAveTime, NumRepl)
Dim i, sec, Min, TimeTot, TimeCol, FitCol, FitTot, NumRows As Integer
Dim TimeIn As String
Dim UppLimTrue As Single, UppLim As Single

   MLECol = GetCol("MLE Time")
   MDECol = GetCol("MDE Time")

   mleTot = 0
   mdeTot = 0
   NumRows = Range("a1").CurrentRegion.Rows.Count
```

```
For i = 2 To NumRows

    MLETimeIn = Cells(i, MLECol).Value
    MLEsec = Val(Right(Left(MLETimeIn, 4), 2))
    MLEMin = Val(Left(MLETimeIn, 1))
    mleTot = mleTot + 60 * MLEMin + MLEsec

    MDETimeIn = Cells(i, MDECol).Value
    MDEsec = Val(Right(Left(MDETimeIn, 4), 2))
    MDEMin = Val(Left(MDETimeIn, 1))
    mdeTot = mdeTot + 60 * MDEMin + MDEsec


Next i

NumRepl = NumRows - 1
MLEAveTime = mleTot / NumRepl
MDEAveTime = mdeTot / NumRepl

End Sub

Function GetCol(StringIn)
  For i = 1 To 256
    If Cells(1, i) = StringIn Then
      GetCol = i
      Exit For
    End If
  Next i
End Function

'
' SummaryHeader Macro
' Macro recorded 1/22/98 by Dean Boerrigter
'
Sub SummaryHeader()
  Range("A1").Formula = "Distribution"
  Range("b1") = "%MDE Better"
  Range("C1") = "MLE Ave Dist"
  Range("D1") = "MLE Std Dist"
  Range("E1") = "MDE Ave Dist"
  Range("F1") = "MDE Std Dist"
  Range("G1") = "MLE Ave Time (s)"
  Range("H1") = "MDE Ave Time (s)"
  Range("I1") = "Replication"
  Range("J1") = "Wsht"
  Range("B2").Select
  ActiveWindow.FreezePanes = True
End Sub
```

Check Mod

```
' This checks for completeness
'
Sub Check()
Dim hold As String

   Range("a1").Sort Key1:=Range("A2"), Order1:=xlAscending, Key2:=Range _
      ("B2"), Order2:=xlAscending, Key3:=Range("J2"), Order3:=xlDescending, Header:=xlGuess,
OrderCustom:=1, _
      MatchCase:=False, Orientation:=xlTopToBottom

   hold = Range("a2").Value
   For i = 2 To Range("a1").CurrentRegion.Rows.Count
      If Cells(i, 1) <> hold Then
         hold = Cells(i, 1)
         Range(Cells(i - 1, 1), Cells(i - 1, 12)).Select
         Call BorderLines
      End If

      If Cells(i, 2).Value = Cells(i - 1, 2).Value Then
         Cells(i, 12) = "*"
      End If
   Next i
End Sub

'
' Puts a border beneath selection
'
Sub BorderLines()
   With Selection.Borders(xlBottom)
      .Weight = xlThin
      .ColorIndex = xlAutomatic
   End With
End Sub
```

Print Mod

```
'
' Setups up Summary Sheet for Printing.
'
Sub setupPrintDriver()
   Call ReplaceSingleDistribution
   Call SplitVariates
   Call SetupPrintSummary
End Sub
'
' SetupPrintSummary Macro
' Macro recorded 2/23/98 by Dean Boerrigter
'
'
Sub SetupPrintSummary()

   Columns("A:K").EntireColumn.AutoFit

   Range("C2:C400").NumberFormat = "0.00%"

   Range("D2:G400").NumberFormat = "0.00000"
'   Range("E2:E400").NumberFormat = "0.00000"
'   Range("F2:F400").NumberFormat = "0.00000"

   Columns("H:I").EntireColumn.Hidden = True

   With ActiveSheet.PageSetup
      .LeftHeader = "&D"
      .CenterHeader = "&A"
      .RightHeader = "&F"
      .LeftFooter = ""
      .CenterFooter = ""
      .RightFooter = ""

      .FitToPagesWide = 1
      .FitToPagesTall = 1
   End With
End Sub
'
' Macro2 Macro
' Macro recorded 2/23/98 by Dean Boerrigter
'
'
Sub ReplaceSingleDistribution()

   Columns("A:A").Select
   Selection.Replace What:=", 99999,0,0,0, 1)", Replacement:=")", _
      LookAt:=xlPart, SearchOrder:=xlByRows, MatchCase:=False
End Sub
```

```
'
' SplitVariates Macro
' Macro recorded 2/24/98 by Dean Boerrigter
'
'
Sub SplitVariates()
   Range("B1").Select
   Selection.EntireColumn.Insert

   Range("A1").CurrentRegion.Select
   Selection.TextToColumns Destination:=Range("A1"), DataType:= _
      xlDelimited, TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter _
      :=False, Tab:=True, Semicolon:=False, Comma:=False, Space _
      :=False, Other:=True, OtherChar:="=", FieldInfo:=Array(Array( _
      1, 1), Array(2, 1))

   Selection.Replace What:="n", Replacement:="", LookAt:=xlPart, _
      SearchOrder:=xlByRows, MatchCase:=False

   Range("B1").FormulaR1C1 = "Variates"

End Sub
```

Open Files Mod

```
Public Const pvDir = "C:\AFIT\Data\"
Public Const pvFileString = "MEFeb08"


'
' OpenFiles Macro
' Macro recorded 1/24/98 by Dean Boerrigter
'
'

Sub OpenFiles()
Dim FileName As String
On Error Resume Next
  For i = 1 To 26
    FileName = pvFileString & Chr(CharCode:=i + 96) & ".xls"

    Workbooks.Open FileName:=pvDir & FileName
'   Call CopyToSummaryWorkbook
'   Windows(FileName).Activate
'   If ActiveWorkbook.Name <> ThisWorkbook.Name Then
'      ActiveWorkbook.Close
'   End If
  Next i

 ' Windows("Summarize Code.xls").Activate
End Sub

'
' CopyToSummaryWorkbook Macro
' Macro recorded 1/22/98 by Dean Boerrigter
'
'

Sub CopyToSummaryWorkbook()
Dim nameIt As String
Dim SummaryWbk As Object


   nameIt = ActiveWorkbook.Name
   Sheets("Output").Copy after:=ThisWorkbook.Sheets(1)


  ActiveSheet.Name = nameIt
End Sub
```

```
' Renames the sheets
'
Sub RenameSheets()
Dim wsht As Object
Dim nameIt As String

nameIt = Application.InputBox("Rename sheets")

i = 0
For Each wsht In ActiveWorkbook.Worksheets
   wsht.Select
   If wsht.Name <> "Summary" Then
      i = i + 1
      wsht.Name = nameIt & i
   End If
Next wsht

End Sub
```

Bibliography

1. Benton-Santo, R. Nicole. Minimum Distance of Mixture Proportions. MS thesis, AFIT/GOR/MA/86D-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1986.
2. Beran, Rudolf. "Minimum Hellinger Distance Estimates for Parametric Models," The Annals of Statistics, Vol. 5, No.3: 445-463 (1977).
3. Bernardo, J.M. "Algorithm AS 103 Psi (Digamma) Function," Applied Statistics Vol. 25, No. 3:315-317 (1976).
4. Burden, Richard L, and J. Douglas Faires. Numerical Analysis 5th Ed. Boston: PWS Publishing Co., 1993.
5. Carroll, David L. "FORTRAN Genetic Algorithm." WWWeb, http://www.staff.uiuc.edu/~carroll/ga.html 6 Jan 1998.
6. Cheng, R. C. H. "The Generation of Gamma Variates with Non-integral Shape Parameter," Applied Statistics Vol. 26 No. 1: 71-75 (1977).
7. Chukwu, Walford and Devendra Gupta. "On Mixing generalized Poisson with generalized Gamma distribution," Metron, Vol. 47: 313-320 (1989).
8. Davis, Lawrence. Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991.
9. Devore, Jay L. Probability and Statistics for Engineering and the Sciences 3rd Ed. Pacific Grove, CA: Brooks/Cole Publishing, 1991.
10. Devroye, Luc. Non-Uniform Random Variate Generation. New York: Springer-Verlag, 1986.
11. Everitt, B.S. and Hand, D.J. Finite Mixture Distributions. London: Chapman and Hall, 1981.
12. Farewell, Vern T. and Ross J. Prentice. "A Study of Distributional Shape in Life Testing," Technometrics, Vol. 19, No. 1: 69-75 (February 1977).
13. Fishman, George S. "Sampling form the Gamma Distribution on a Computer," Communications of the ACM Vol. 19 No. 6: 407-409 (July 1976).
14. Gallagher, Mark A. and Albert H. Moore. "Robust Minimum Distance Estimation Using the 3-Parameter Weibull Distribution".
15. Goldberg, David E. Genetic Algorithms in Search, Optimization, and Machine Learning Reading, Reading, MA: Addison-Wesley, 1989.
16. Goldberg, David. "FTP simple GA code." Pascal code from Goldberg, D. E. (1989), "Genetic Algorithms in Search, Optimization, and Machine Learning." WWWeb, ftp://ftp-illigal.ge.uiuc.edu/pub/src/simpleGA/Pascal/ 17 Jan 1998.
17. Grefenstette, John J. "Optimization of Control Parameters for Genetic Algorithms," in Genetic Algorithms. Ed. Bill Buckles and Frederick Petry. Los Alamitos, CA: IEEE Computer Society Press, 1992.
18. Hager, Harold W. and Lee J.Bain. "Inferential Procedures for the Generalized Gamma Distribution," Journal Of The American Statistical Association Vol. 65 No. 332: 1601-1609 (December 1970).
19. Hager, Harold W, Lee J. Bain and Charles E. Antle. "Reliability Estimation for the Generalized Gamma Distribution and Robustness of the Weibull Model," Technometrics Vol. 13, No. 3: 547-557 (August 1971).

20. Harter, H. Leon. Asymptotic Variances and Covariances of Maximum-Likelihood Estimators, from Censored Samples, of the Parameters of a Four-Parameter Generalized Gamma Distribution. Wright-Patterson AFB: Aerospace Research Laboratories, Office of Aerospace Research, August 1966 (ARL 66-0158).

21. Harter, H. Leon "Maximum-Likelihood Estimation of the Parameters of a Four-Parameter Generalized Gamma Population from Complete and Censored Data," Technometrics, Vol. 9, No. 1: 159-165 (February 1967).

22. Hobbs, Jon, Albert Moore and William James "Minimum Distance Estimation of the Three Parameters of the Gamma Distribution," IEEE Transactions on Reliability Vol. R-33 No. 3: 237-240 (August 1984).

23. Holland, John H. Adapation in Natural and Artificial Systems Ann Arbor: The University of Michigan Press, 1975.

24. Holland, John. Hidden Order. Reading, MA: Addison-Wesley Publishing, 1995.

25. J. Wolfowitz. "The Minimum Distance Method," Annals of Mathematical Statistics Vol. 28: 75-88 (1957).

26. James, William L. Robust Minimum Distance Estimation Based on a Family of Three-Parameter Gamma Distributions MS thesis, AFIT/GOR/MA/80D-4. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1980.

27. Jonk, M.D. "Erzeugung von Betaverteilen und Gammaverteilen Zufallszahlen (in German)," Metrika Vol. 8: 5-15 (1964).

28. Kinderman, A.I. and J.F. Monahan. "New Methods for Generating Student's t and Gamma Variates," Computing Vol. 25: 369-377 (1980).

29. Krishnakumar, Kalmanje. "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," SPIE Intelligent Control and Adaptive Systems, Vol. 1196: 289-296 (1989).

30. Kuester, James L and Joe H. Mize. Optimization Techniques with Fortran, New York: McGraw-Hill,1973.

31. Law, Averill M. and W. David Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill, 1991.

32. Lawless, J. F. Statistical Models and Methods for Lifetime Data. New York: John Wiley & Sons, 1982.

33. Leemis, Lawrence and Bruce Schmeiser. "Random Variate Generation for Monte Carlo Experiments" IEEE Transactions on Reliability Vol. R-34 No. 1: 81-85 (April 1985).

34. Lienhard, John H. and Meyer, Paul L. "A Physical Basis for the Generalized Gamma Distribution," ?? Vol. 25, No. 3: 330-334 (1964).

35. Marsalglia, G. "The Squeeze Method For Generating Gamma Variates," Computing and Mathematics with Applications Vol. 3: 321-325 (1977).

36. Mendenhall, William, Wackerly, Dennis D. and Scheaffer, Richard L. Mathematical Statistics with Applications. (4th Edition) Boston: PWS-Kent, 1990.

37. Mumford, D. "Robust Parameter Estimation for the Mixed Weibull(Seven Parameter) Including the Method of Maximum Likelihood and the Method of Minimum Distance." MS thesis, AFIT/GOR/ENY/97M-1. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1997.

38. Özmen, Tamir. A Modified Anderson Darling Goodness-of-Fit for the Gamma Distribution with Unknown Scale and Location Parameters. MS thesis, AFIT/GOR/ENS/93M-24. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993.

39. Parr, Van B. and J.T. Webster. "A Method for Discriminating Between Failure Density Functions Used in Reliability Predictions," Technometrics Vol. 7 No.1: 1-10 (February 1965).

40. Parr, William C. and William R Schucany "Minimum Distance and Robust Estimation," Journal Of The American Statistical Association (JASA) Vol. 75, No. 3, 616-624 (1980).

41. Parrish, Donald A. "Genetic Algorithm Approach to Automating Satellite Range Scheduling" MS thesis, AFIT/GOR/ENS/94M-10. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1994.

42. Pham, T and J Almhana. "The Generalized Gamma Distribution: Its Hazard Rate and Stress-Strength Model," IEEE Transactions on Reliability Vol. 44, No. 3: 392-397 (September 1995).

43. Radhakrishna, C. A.V. Datattreya Rao and G.V.S.R. Anjaneyulu. "Estimation of Parameters in a Two-Component Mixture Generalized Gamma Distribution," Communication in Statistics-Theory Methods Vol. 21 No. 6: 1799-1805 (1991).

44. Rao, A.V. and R.R.L Kantam and V. L. Narasimham. "Linear Estimation of Location and Scale Parameters in the Generalized Gamma Distribution," Communications in Statistics: Theory and Methods. Vol. 20, No. 12: 3823-3848 (1991).

45. Rawlins, Gregory. Foundations of Genetic Algorithms. San Mateo, CA: Morgan Kaufmann Publishers, 1991.

46. Redner, R.A. MLE For Mixture Models. JSC-1632, 1980.

47. Sarkar, T.K and Vignaux, G.A. "A Note on Generalized Gamma Distribution and Reliability," Pacific Statistical Congress: 280-282 (1985).

48. Shammas, Namir C. Mathematical Algorithms in Visual Basic for Scientists & Engineers. New York: McGraw-Hill, 1996.

49. Shumaker, Keith. Robust Parameter Estimation for the Four-Parameter Generalized Gamma Distribution. MS thesis, AFIT/LSH. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, September 1982 (ADA 124077).

50. Sinclair, C.D., B. D. Spur and M. I. Ahmad. "Modified Anderson Darling Test," Commun. Statist-Theory Meth. Vol. 19 No. 10: 3677-3686 (1990).

51. Stacy, E.W. "A Generalization of the Gamma Distribution," Ann. Math. Statist., Vol. 33: 1187-1192 (1962).

52. Stacy, E.W. and Mihram G. A. "Parameter Estimation for a Generalized Gamma Distribution," Technometrics, Vol. 7 No. 3: 349-358 (1965).

53. StatLab. "AS 103." Code for the Psi or Digamma Function in FORTRAN, WWWeb, http://www.statlab.uni-heidelberg.de/mirrors/statlib/apstat/103. 20 Jan 1998.

54. Stephens, M.A. "EDF Statistics for Goodness of Fit and Some Comparisons," JASA Vol. 69, No. 347: 730-737 (September 1974).

55. Tadikamalla, Pandu. "Computer Generation of Gamma Random Variables," Communications of the ACM Vol. 21 No. 5: 419-422 (May 1978).

56. Tadikamalla, Pandu. "Computer Generation of Gamma Random Variables-II," Communications of the ACM Vol. 21 No. 11: 925-928 (November 1978).

57. Tadikamalla, P.R. and M.E. Johnson. "A Survey of Computer Methods for Sampling from the Gamma Distribution," Proceedings of the 1978 Winter Simulation Conference: 131-135 (Forthcoming 1978).

58. Tadikamalla, P.R. "Random Sampling from the Generalized Gamma Distribution," Computing Vol. 23: 199-203 (1979).

59. Tadikamalla, Pandu and Mark E. Johnson. "A Complete Guide to Gamma Variate Generation," American Journal of Mathematical and Management Sciences Vol. 1, No. 3: 213-236 (1981).

60. Tan, W.Y. and W.C. Chang. "Some Comparisons of the Method of Moments and the Method of Maximum Likelihood in Estimating Parameters of a Mixture of Two Normal Densities," JASA Theory and Methods, Vol. 67: 702-708 (September 1972).

61. Titterington, D.M.,Smith, A.F.M and Makov, U.E. Statistical Analysis of Finite Mixture Distributions, Chichester, John Wiley & Sons, 1985.

62. Wallace, N.D. "Computer Generation of Gamma Random Variates with Non-integral Shape Parameters," Communications of the ACM Vol. 17, No. 12: 691-695 (December 1974).

63. Whittaker, J. "Generating Gamma and Beta Random Variates with Non-integral Shape Parameters," Applied Statistics Vol. 23 No. 2: 210-214 (1974).

64. Wingo, Dallas. "Computing Maximum-Likelihood Parameter Estimates of the Generalized Gamma Distribution by Numerical Root Isolation" IEEE Transactions on Reliability Vol. R-36 No. 5: 586-590 (December 1987).

Vita

Captain Dean Boerrigter was born 21May 1970 in Washington, D. C. He graduated from Oxon Hill High School in 1988, at which time he entered undergraduate studies at the U.S. Air Force Academy in Colorado. He graduated with a Bachelor of Science degree in Operations Research in May 1992. He received his Air Force commission upon graduation.

His first assignment was to the Air Force Wartime Manpower and Personnel Readiness Team (AFWMPRT) at Fort Ritchie, MD where he worked as a Wartime Analyst for the Air Staff. In August 1996, he entered the School of Engineering, Air Force Institute of Technology. His follow-on assignment was to AFOTEC at Kirtland AFB, NM.

Permanent Address: 1665 Elkhorn Dr

Arlington, NE 68002

wvqr71a@prodigy.com

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March, 1998

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Parameter Estimation Of The Mixed Generalized Gamma Distribution Using Maximum Likelihood Estimation And Minimum Distance Estimation

5. FUNDING NUMBERS

6. AUTHOR(S)

Dean G. Boerrigter, Capt., USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
2750 P. Street
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Advisors: Maj Ed Pohl, epohl@afit.af.mil ; Maj William P. Murdock, wmurdock@afit.af.mil DSN 225-3636

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Generalized Gamma is an extremely flexible distribution that is useful for reliability modeling. Among its many special cases are the Weibull and Exponential distributions. A mixture of Generalized Gamma Distributions is even more useful because multiple causes of failure can be simultaneously modeled. This research studied parameter estimation of the special cases of the Mixed Generalized Gamma Distribution and built upon them until the full nine-parameter distribution was being estimated.

Two techniques were used to estimate the parameters of each distribution. The first technique used was the Method of Maximum Likelihood. The log likelihood equation was maximized using a Genetic Algorithm. The second technique used was the Method of Minimum Distance. This technique takes the Maximum Likelihood parameter estimate as initial estimate. With this initial estimate, the mixture and the first location parameter are sequentially varied to minimize the Anderson-Darling statistic between the estimated cumulative distribution function and the empirical distribution function. These two parameters are then fixed at their Minimum Distance values and the remaining parameters are re-estimated using Maximum Likelihood.

Minimum Distance Estimation was demonstrated to improve the parameter estimates from Maximum Likelihood for almost all of the special case distributions tested. It did not improve the estimate for the full nine-parameter Mixed Generalized Gamma Distribution, but this was because the technique used to find the Maximum Likelihood parameter estimates performed poorly and did not return a good initial estimate for Minimum Distance.

14. SUBJECT TERMS
Generalized Gamma, Maximum Likelihood, Minimum Distance, Genetic Algorithm, Parameter Estimation, Micro-Genetic Algorithm

15. NUMBER OF PAGES
134

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL